Contents

vii Authors
ix Conference Committee
xiii Introduction to the Invited Panel Discussion
xv Invited Panel Slides

SESSION 1  MULTISENSOR FUSION, MULTITARGET TRACKING, AND RESOURCE MANAGEMENT I

10646 02 Track stitching and approximate track association on a pairwise-likelihood graph [10646-1]
10646 03 From labels to tracks: it's complicated [10646-2]
10646 04 An introduction to the generalized labeled multi-Bernoulli filter through Matlab code [10646-3]
10646 05 On-orbit calibration of satellite based imaging sensors [10646-5]
10646 06 The data-driven δ-generalized labeled multi-Bernoulli tracker for automatic birth initialization [10646-4]

SESSION 2  MULTISENSOR FUSION, MULTITARGET TRACKING, AND RESOURCE MANAGEMENT II

10646 07 Localization of a point target from an optical sensor’s focal plane array [10646-6]
10646 09 Trajectory estimation and impact point prediction of a ballistic object from a single fixed passive sensor [10646-8]
10646 0A A study of particle filtering approaches for the kidnapped robot problem [10646-9]
10646 0B Evaluation of optimizations of Murty’s M-best assignment [10646-10]

SESSION 3  INFORMATION FUSION METHODOLOGIES AND APPLICATIONS I

10646 0C A generalized labeled multi-Bernoulli filter for correlated multitarget systems [10646-11]
10646 0D A clutter-agnostic generalized labeled multi-Bernoulli filter [10646-12]
SESSION 4  INFORMATION FUSION METHODOLOGIES AND APPLICATIONS II

10646 0G  Detection system fusion based on the predictive value curve and its variations [10646-15]
10646 0H  Improving ATR system performance through sequences of classification tasks [10646-16]

SESSION 5  INFORMATION FUSION METHODOLOGIES AND APPLICATIONS III

10646 0K  pystemlib: towards an open-source tracking, state estimation, and mapping toolbox in Python [10646-19]
10646 0L  Analysis of noise impact on distributed average consensus [10646-20]

SESSION 6  INFORMATION FUSION METHODOLOGIES AND APPLICATIONS IV

10646 0M  A framework for adaptive MaxEnt modeling within distributed sensors and decision fusion for robust target detection/recognition [10646-21]
10646 0N  A comparison between robust information theoretic estimator and asymptotic maximum likelihood estimator for misspecified model [10646-22]
10646 0O  Optimizing collaborative computations for scalable distributed inference in large graphs [10646-23]
10646 0P  Enabling self-configuration of fusion networks via scalable opportunistic sensor calibration [10646-24]
10646 0Q  Multiscale graph-based framework for efficient multi-sensor integration and event detection [10646-25]

SESSION 7  INFORMATION FUSION METHODOLOGIES AND APPLICATIONS V

10646 0S  An adaptive sensing approach for the detection of small UAV: first investigation of static sensor network and moving sensor platform [10646-27]
10646 0U  MARINE-EO bridging innovative downstream earth observation and Copernicus enabled services for integrated maritime environment, surveillance, and security [10646-29]
10646 0V  A novel architecture for behavior/event detection in security and safety management systems [10646-30]
<table>
<thead>
<tr>
<th>SESSION 8</th>
<th>SIGNAL AND IMAGE PROCESSING, AND INFORMATION FUSION APPLICATIONS I</th>
</tr>
</thead>
<tbody>
<tr>
<td>10646 0W</td>
<td>A new FSII-CFAR detector based on fuzzy membership degree [10646-31]</td>
</tr>
<tr>
<td>10646 0Z</td>
<td>Low-cost multi-camera module and processor array for the ultra-fast framerate recognition, location, and characterization of explosive events [10646-34]</td>
</tr>
<tr>
<td>10646 10</td>
<td>Super-resolution of remote sensing images using edge-directed radial basis functions [10646-35]</td>
</tr>
<tr>
<td>10646 11</td>
<td>Compressive sensing using the log-barrier algorithm with complex-valued measurements [10646-36]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SESSION 9</th>
<th>SIGNAL AND IMAGE PROCESSING, AND INFORMATION FUSION APPLICATIONS II</th>
</tr>
</thead>
<tbody>
<tr>
<td>10646 12</td>
<td>Stabilization and registration of full-motion video data using deep convolutional neural networks [10646-37]</td>
</tr>
<tr>
<td>10646 13</td>
<td>iSight: computer vision based system to assist low vision [10646-38]</td>
</tr>
<tr>
<td>10646 14</td>
<td>A real-time object detection framework for aerial imagery using deep neural networks and synthetic training images [10646-39]</td>
</tr>
<tr>
<td>10646 15</td>
<td>Mobile crowd-sensing for access point localization [10646-40]</td>
</tr>
<tr>
<td>10646 16</td>
<td>Going deeper with CNN in malicious crowd event classification [10646-41]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SESSION 10</th>
<th>SIGNAL AND IMAGE PROCESSING, AND INFORMATION FUSION APPLICATIONS III</th>
</tr>
</thead>
<tbody>
<tr>
<td>10646 1A</td>
<td>Scanning LiDAR for airfield damage assessment [10646-47]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SESSION 11</th>
<th>SIGNAL AND DATA PROCESSING FOR SMALL TARGETS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10646 1F</td>
<td>Robust spectral classification [10646-50]</td>
</tr>
<tr>
<td>10646 1H</td>
<td>Separation of small targets in multi-wavelength mixtures based on statistical independence [10646-52]</td>
</tr>
<tr>
<td>10646 1K</td>
<td>Error statistics of bias-naive filtering in the presence of bias [10646-55]</td>
</tr>
<tr>
<td>10646 1L</td>
<td>An analytic solution to ellipsoid intersections for multistatic radar [10646-56]</td>
</tr>
</tbody>
</table>
Multilevel probabilistic target identification methodology utilizing multiple heterogeneous sensors providing various levels of target characteristics

POSTER SESSION

Attitude control system for a balloon based telescope

Blind modulation detection via staged GLRT

CRLB for estimation of 3D sensor biases in spherical coordinates
Authors

Numbers in the index correspond to the last two digits of the seven-digit citation identifier (CID) article numbering system used in Proceedings of SPIE. The first five digits reflect the volume number. Base 36 numbering is employed for the last two digits and indicates the order of articles within the volume. Numbers start with 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B...0Z, followed by 10-1Z, 20-2Z, etc.

Altman, Emilie, 0K
Anderson, Evan, 1K
Angott, Bill, 1A
Astyakopoulos, Alkis, 0U
Bacher, E., 0S
Bailey, Colleen P., 1R
Balki, Bhashyam, 0K
Balasingam, Balakumar, 07
Bar-Shalom, Yaakov, 05, 07, 09, 0B, 1S
Bayer, Michael A., 12
Belafdel, Djedjiga, 05, 1S
Ben-Dov, R., 09
Borel-Donohue, Christoph, 14
Bothos, John, 0V
Carniglia, Peter, 0K
Chance, Zachary, 1K
Chandran, Krishnan, 13
Chen, Lingji, 02, 03, 04
Colonna-Romano, John, 0O
Copsey, Keith, 0P
Couwenhoven, Doug W., 12
Cox, Kevin, 0Z
da Silva, Felipe B., 11
DeMars, Kyle J., 06
Diltz, Robert, 1A
Dou, Wenbo, 0B
Dunham, Joel, 1M
Emge, Darren K., 1H
English, Woody, 1A
Eun, Sungmin, 16
Fowler, Stuart, 1A
Gatsak, Tatiana, 0K
Giovannetchi, F., 0S
Gómez Miguel, Beatriz, 0U
Grewe, Lynne, 13
Hammer, M., 0S
Hengy, S., 0S
Hommes, A., 0S
Hurley, Jeffery D., 1M
Johannes, W., 0S
Johnson, Clint, 1M
Jones, Brandon A., 0F
Kadar, Ivan, 0M
Kashyap, Archana, 13
Kay, Steven M., 0N, 1F
Kong, Yingying, 0W
Kontoes, Haris, 0U
Kowalski, Michael, 1S
Kwon, Heesung, 14, 16
Kyriazanos, Dimitris M., 0U, 0V
Langford, Darrell, 1A
Laurenzis, M., 0S
Lee, Hyungtae, 14, 16
LeGrand, Keith A., 06
Leirens, Sylvain, 15
Le Moigne, Jaqueline, 10
Leung, Henry, 0L, 0W
Levchuk, Georgiy, 0O
Li, Boyuan, 0L
Lu, Qin, 07, 0B
Lykousis, Vasiliou, 0U
Mahler, Ronald, 0C, 0D, 0E
Maraviglia, Carlos, 0Z
Mareboyana, Manohar, 10
Markellou, Marina, 0U
Martins, Ana, 0U
McArdle, Sean M., 0F
McGrom, B., 09
Mioso, Cristiano J., 11
Mohler, David, 0A
Mowakeaa, Rami, 1H
Muligrew, Bernard, 0P
Narayanan, Priya, 14
Nehmetallah, George, 1P
Oxley, Mark E., 0G, 0H
Pados, Dimitris A., 1R
Page, Scott, 0P
Park, Sungjoo, 0Z
Pattipati, Krishna, 0B
Pereira da Silva, Alex, 15
Poyet, J.-M., 0S
Prasad, Lakshman, 0Q
Rao, Raghuveer, 14
Rassy, O., 0S
Ravago, Nicholas, 0F
Relyea, Stephen, 1K
Rinehart, Stephen A., 1P
Rizogiannis, Constantinou, 0U, 0V
Robinson, Brian, 1A
Rumbley, Sarah E., 02
Scheitzer, S., 0S
Schubert Kabbab, Christine M., 0G, 0H
Seneviratne, Chatura, 0L
Shah, Akhil K., 0F
Shahshahani, Allen, 13
Shahshahani, Jake, 13
Shapero, Samuel A., 1L
Simmons, Jimmy, 1M
Conference Committee

Symposium Chair
Arthur A. Morrish, Raytheon Space and Airborne Systems (United States)

Symposium Co-chair
Ruth L. Moser, Air Force Research Laboratory (United States)

Conference Chairs
Ivan Kadar, Interlink Systems Sciences, Inc. (United States)

Conference Co-chairs
Bhashyam Balaji, Defence Research and Development Canada (Canada)
Erik P. Blasch, Air Force Research Laboratory (United States)
Lynne L. Grewe, California State University, East Bay (United States)
Thia Kirubarajan, McMaster University (Canada)
Ronald P. S. Mahler, Random Sets, LLC (United States)

Conference Program Committee
Mark G. Alford, Air Force Research Laboratory (United States)
William D. Blair, Georgia Tech Research Institute (United States)
Mark J. Carlotto, General Dynamics Advanced Information Systems (United States)
Alex L. Chan, U.S. Army Research Laboratory (United States)
Kuo-Chu Chang, George Mason University (United States)
Chee-Yee Chong, Independent Consultant (United States)
Marvin N. Cohen, Georgia Tech Research Institute (United States)
Frederick E. Daum, Raytheon Company (United States)
Jean Dezert, Office National d’Etudes et de Recherches Aérospatiales (France)
Mohammad Farooq, AA Scientific Consultants Inc. (Canada)
Laurie H. Fenstermacher, Air Force Research Laboratory (United States)
Charles W. Glover, Oak Ridge National Laboratory (United States)
I. R. Goodman, Consultant (United States)
Michael L. Hinman, Independent Consultant (United States)
Jon S. Jones, Air Force Research Laboratory (United States)
Georgiy M. Levchuk, Aptima, Inc. (United States)
Session Chairs

1. Multisensor Fusion, Multitarget Tracking, and Resource Management I
   Ivan Kadar, Interlink Systems Sciences, Inc. (United States)

2. Multisensor Fusion, Multitarget Tracking, and Resource Management II
   Ivan Kadar, Interlink Systems Sciences, Inc. (United States)

3. Information Fusion Methodologies and Applications I
   Ronald P. S. Mahler, Random Sets LLC (United States)

4. Information Fusion Methodologies and Applications II
   Ivan Kadar, Interlink Systems Sciences, Inc. (United States)
   Chee-Yee Chong, Independent Consultant (United States)

5. Information Fusion Methodologies and Applications III
   Chee-Yee Chong, Independent Consultant (United States)
   Ivan Kadar, Interlink Systems Sciences, Inc. (United States)

6. Information Fusion Methodologies and Applications IV
   Chee-Yee Chong, Independent Consultant (United States)
   Ivan Kadar, Interlink Systems Sciences, Inc. (United States)

7. Information Fusion Methodologies and Applications V
   Erik Blasch, Air Force Research Laboratory (United States)
   Ivan Kadar, Interlink Systems Sciences, Inc. (United States)
   Chee-Yee Chong, Independent Consultant (United States)
8 Signal and Image Processing, and Information Fusion Applications I
Lynne L. Grewe, California State University, East Bay (United States)
Alex L. Chan, U.S. Army Research Laboratory (United States)

9 Signal and Image Processing, and Information Fusion Applications II
Lynne L. Grewe, California State University, East Bay (United States)
Alex L. Chan, U.S. Army Research Laboratory (United States)

10 Signal and Image Processing, and Information Fusion Applications III
Alex L. Chan, U.S. Army Research Laboratory (United States)
Lynne L. Grewe, California State University, East Bay (United States)

11 Signal and Data Processing for Small Targets
Philip D. West, Georgia Tech Research Institute (United States)
Darren K. Emge, U.S. Army Edgewood Chemical Biological Center (United States)
Introduction to the Invited Panel Discussion

Deep Learning in AI and Information Fusion

In the early days of artificial intelligence (AI) starting, say in the 1970s and 1980s, the predominant reasoning methods were logical and symbolic, using, e.g., Lisp/Prolog languages. Later in the 1980s, AI tools were used such as Knowledge Environment Engineering (KEE) and Automated Reasoning Tool (ART) expert systems, and early heuristic reasoning methods. Also, the concept and mathematical representation of “context” logic was defined. The concept and apps of both “knowledge based” and “context” are currently used in several apps in information fusion (IF) along with several methods to apply and learn contextual information.

In the early 1980’s, AI was viewed as the solution to information fusion problems. In fact, many contributors to the first distributed sensor networks program were AI researchers. However, inadequate computing and AI approaches such as expert systems and heuristic uncertainty reasoning could not address the challenges of information fusion. Thus, important advances in information fusion, and in particular, multi-target tracking, were made with little contribution from AI.

During the long AI winter, researchers addressed the deficiencies of early AI, developing rigorous representation and reasoning techniques for uncertainty, and machine learning approaches. Recently, data science was established as a popular area to exploit the large volumes of data (a.k.a. Big Data) collected by physical sensors and online activities using machine learning and other analytic tools.

Artificial intelligence and data science pose both challenges and opportunities to IF. They are challenges because they appear to address the same problems as information fusion, but with more powerful techniques, thus siphoning away both research funding and research talent. However, these challenges can also be opportunities because AI and data science provide new research directions for information fusion. Examples include: IF with big data, hard and soft data fusion, learning about context, graph techniques for tracking and fusion, dynamic network analysis, apps to cyber and imagery processing.

The objective of this panel was to bring to the attention of the fusion community the importance of the application of deep learning in AI and IF, highlighting issues, illustrating approaches and addressing challenges. A number of invited experts discussed challenges in processing and research, and addressed these challenges with IF. The panelists illustrated parts of the above-mentioned areas over different applications and association with IF. The panel highlighted impending issues and challenges using conceptual and real-world related examples associated with the applications of above.

Chee-Yee Chong
Ivan Kadar
Invited Panel Discussion
Deep Learning in AI and Information Fusion

Organizers
Chee-Yee Chong, Independent Consultant
Ivan Kadar, Interlink Systems Sciences, Inc.
Erik Blasch, Air Force Research Lab

Moderators
Ivan Kadar, Interlink Systems Sciences, Inc.
Chee-Yee Chong, Independent Consultant

April 16, 2018
SPIE Conference 10646
“Signal Processing, Sensor Fusion and Target Recognition XXVII”
Orlando, FL 16-19 April 2018

Invited Panel Discussion

Panel Participants:
Dr. Erik Blasch, Air Force Research Lab., U.S.A.
Dr. Chee-Yee Chong, Independent Consultant, U.S.A.
Professor George Cybenko, Dartmouth College, NH, U.S.A
(unable to attend
Professor Lynne Grewe, California State Univ, East Bay, U.S.A
Dr. Ivan Kadar, Interlink Systems Sciences, Inc., U.S.A.
Dr. Uttam K. Majumber, Air Force Research Lab., U.S.A
Invited Panel Discussion

Presentation Topics

“Challenges of Using Deep Learning for Trusted Sensor Fusion”
Dr. Chee-Yee Chong, Independent Consultant

“Deep Learning and Computer Vision: Guidelines and Special Issues”
Professor Lynne Grewe and Garrett Stevenson, California State Univ, East Bay, CA

“Deep Learning for Object Recognition from High Volume Radio Frequency Data”
Dr. Uttam K. Majumber, Air Force Research Lab.

“Retrospectives on the Application of AI & Deep Learning in Information Fusion” (addendum to presentations)
Dr. Ivan Kadar, Interlink Systems Sciences, Inc.
Challenges of Using Deep Learning for Trusted Information Fusion

Chee-Yee Chong
Independent Researcher
Los Altos, USA
cychong@ieee.org

Presented at Panel on Deep Learning in AI and Information Fusion
SPIE Signal Processing, Sensor/Information Fusion, and Target Recognition XXVII
Orlando, Florida
April 16, 2018

Outline

• History of AI for information fusion
• Deep learning benefits and issues
• Framework for trusted information fusion
Artificial Intelligence (AI) Is Everywhere

• AI has recent spectacular successes defeating humans
  • Deep Blue beat world champion in chess (1997)
  • Watson won US Jeopardy quiz show (2011)
  • AlphaGo beat world’s top player (Ke Jie) in go (2017)
• AI is in everyday applications
  • Smart phones – speech recognition, machine translation
  • Homes – smart thermostat, robotic vacuum
  • Cars – driver assistance
• In particular, AI/deep learning is used in sensor, data and information fusion, e.g.,
  • BBC news, 4/13/2018: Chinese police have used facial recognition technology to locate and arrest a man who was among a crowd of 60,000 concert goers.

Artificial Intelligence (AI) Poses Challenges to Information Fusion

• AI is viewed as the solution for all fusion problems
  • Low-level processing and object recognition
  • Video surveillance
  • Activity detection and behavioral analysis
  • Network and patterns of life analysis
• AI/deep learning is more visible due to
  • Beating humans in many applications
  • Widely available hardware, software, and training data for development
  • Successful use in many systems
• Results
  • Sponsors turn to AI to solve information fusion problems
  • AI attracts more students, researchers, and funding
  • Virtuous cycle (funding -> research -> success -> more funding)
Information Fusion is a Natural Application for AI

- Humans solve information fusion problems all the time
  - Low level perception – environment, objects
  - High level understanding – situation, prediction

AI Approaches for Information Fusion Are Evolving

- Expert systems (mimic human experts)
  - Medical diagnosis
  - Signal understanding
- Probabilistic reasoning (with models)
  - Object recognition
  - Situation understanding
- Neural networks / deep learning
  - Feature extraction
  - Speech understanding
  - Object recognition
  - Video tracking

J. Launchbury, "A DARPA perspective on artificial intelligence," 2017
Expert Systems for Fusion – MYCIN (~1975)

- Medical diagnosis
  - Inputs: test results
  - Outputs: infectious disease
- Rule-based system
  - Knowledge base of few hundred IF_THEN rules: IF symptom A THEN disease B
  - Inference engine by backward chaining
  - Certainty factors to represent uncertainty
  - Heuristic combination of evidence
  - Performs better than many doctors
  - Stimulated research on uncertainty reasoning


Expert Systems for Fusion – HASP/SIAP (1970’s)

- Signal understanding system
  - Inputs: acoustic signals from hydrophones
  - Outputs: detection, location and type of vessel
- Rule-based system
  - Hierarchy of rules for signal to symbol transformation
  - Inference by knowledge sources responsible for different levels of processing

Issues with Expert Systems for Information Fusion

- Knowledge acquisition
  - Finding experts that can articulate their reasoning; experts with good intuition are not suitable
  - Extracting knowledge from experts
- Knowledge representation
  - Consistency and completeness of rules
  - Representation of uncertainty
- Inference engine
  - Control of inference
  - Reasoning with uncertainty
  - Processing speed

Al in 1980’s

- 1981 – Japan started “Fifth Generation Computer” project to build intelligent computers
- United States responded with “Strategic Computing Initiative” with AI as main objective, including Autonomous Land Vehicle (ALV)
- ALV followed road in 1985 demo but vision system was very sensitive to
  - Light and shadow – detect road edge at noon, but not with shadow at dusk
  - Environmental change (like mud left along road by another vehicle)
- Booming AI industry (software, hardware) became a bust with AI winter in late 1980’s

Neural Networks (1980's)

- Motivated by physiology and function of neurons in brain
- Long history
  - McCulloh, Pitts – 1940’s
  - Widrow – 1960’s
  - Hopfield, Rumelhart, Hinton – 1980’s
- Weights learned from training data
- Excellent for low level recognition task
- Implementation issues
  - Black box approach cannot explain results
  - Performance sensitive to training data

Uncertainty Reasoning in AI

- Information fusion has dealt with uncertainty in evidence (input) and knowledge or data
  - Drawback of neural network approach
  - Recognized very early by expert system developers
- Uncertainty reasoning approaches
  - Rule-based methods
  - Probabilistic reasoning
  - Evidence theory
    - Dempster Shafer
    - Dezert-Smarandache Theory (DSmT)
  - Fuzzy sets
- Probabilistic reasoning became very popular in 1980’s

Probabilistic Reasoning/Graphical Models

- Probability model expressed graphically as networks
  - Nodes are random variables
  - Weights on edges represent conditional probabilities
- Inference computes conditional probabilities given evidence
  - Node elimination
  - Junction tree
  - MCMC
- Very natural for researchers with background in estimation theory
- Considered AI because of separation into knowledge and automatic inference

Military Unit Detection from Synthetic Aperture Radar (SAR) Imagery

Model-Based Object Recognition

Model Driven ATR

Image
Focus Attention
Index

On-Line Models
Predict Features
Match Features
Extract Features

Hypothesize & Test

Target Reports

Context & Collated Information

Sensor, Image Formation, and Acquisition Parameters

Image & Hypothesis Space Reduction

Moving and Stationary Target Acquisition and Recognition (MSTAR)

• Program manager said "neural network is not allowed"
AI for Information Fusion from 1990's to 2010

- Probabilistic graphical models become main AI approach for fusion
  - Rigorous treatment of uncertainty
  - Model-based approach is explainable
  - Many Inference techniques
  - Models can be learned from data
  - Can be extended to handle evidence theory, e.g., valuation networks
- Mathematic framework is similar to that of tracking
  - Predict features from model
  - Match extracted features with prediction (association problem)
- Meanwhile, neural networks are used for many low level functions where modeling is difficult and training is easy
- Then computers become more powerful and massive amounts of data are available

Shift from Knowledge-Based AI to Learning-Based AI

- Explicit problem knowledge
- Manual knowledge acquisition and representation
- Transparent fusion processing
- Problem knowledge captured by data
- Training data acquisition without knowledge representation
- Black box fusion processing
Deep Learning

- Deep neural network uses multiple hidden layers between input and output layers to model complex nonlinear relationships
- Input layers can be images or audio signals instead of features

What Makes Deep Learning Possible

- Deep learning is possible due to advances in computing power and available data
Deep Learning is Drawing Attention Away From Traditional Information Fusion

- Almost anyone can apply deep learning for his/her problems
  - Open source software, e.g., TensorFlow, Theano
  - Public domain data, e.g., ImageNet, Open Images Dataset
  - Inexpensive powerful hardware, e.g., Nvidia, Intel, Google
- Deep learning has been successfully used in
  - Video surveillance
  - Object and threat detection
  - Driverless vehicles
  - Cyber security (where modeling is very difficult and data is plentiful)

Challenges of Using Deep Learning for Trusted Information Fusion

- Performance is only as good as data
  - Large amounts of data are needed
  - Training data for rare events are sparse
- Results are hard to explain
  - Black box provides no visibility
  - Research on explainable AI is still ongoing
  - Thus few machine learning systems are used in critical missions or making life death decisions
DARPA Explainable AI Program Objective

Today
Learning Process
Training Data
Learned Function
This is a cat (p = 0.9)
User with a Task

Tomorrow
New Learning Process
Training Data
Explainable Model
Explanation Interface
This is a cat: Has fur, whiskers, and claws.
User with a Task

Framework for Developing Trusted Fusion Systems

- Integrated knowledge and learning approach*
- Inference based on problem knowledge
  - Explicit knowledge representation
  - Explainable results
- Model parameters from machine learning
  - Learning when model knowledge is weak and data are available
  - Testable parameter estimation

* ONR solicitation (N00014-18-R-SN05) on integration of domain knowledge and machine learning to address shortcomings of deep learning (large training sets, brittleness, explainability, rare and complex events, etc.)
**Direct Learning of Sensor Input to Object Tracks**

- Related to PHD
- Training and performance issues


---

**Association Graph for Tracking**

- Association graph provides efficient representation for possible associations
  - Nodes: measurements or tracklets
  - Edges: possible associations
  - Paths: tracks
  - Path cover: association hypothesis
- Track likelihood is sum of pairwise association likelihoods under Markov assumptions
  \[ P(y_{i:t} | y_1, \ldots, y_t) = P(y_{i:t} | y_t) \]
- Best association hypothesis can be computed in polynomial time as
  - Bipartite matching
  - Minimum cost network flow

Learning Association Costs in Association Graph

- Pairwise association costs can be computed
  - Attributes of detections
  - Measurement models
  - Alternatively, they can be learned from data using back propagation


Integrated Knowledge/Learning for Multitarget Tracking

- Direct learning
  - Questionable performance
  - Hard to explain
- Integrated knowledge/learning
  - Higher performance
  - Explainable
Conclusions

• Information fusion researchers have love hate relationship with computers
  • Love to develop algorithms for computers
  • Hate when computers become too smart to take over their job
• Users do not care about specific approach as long as it provides solution
• AI and deep learning still cannot solve all fusion problems, especially in trusted or mission critical systems
• Trusted fusion system should exploit
  • Knowledge when it is available and can be represented for inference
  • Machine learning when knowledge is weak and data are available
Deep Learning in AI and Information Fusion Panel

Computer Vision Today & Deep Learning

Computer Vision is becoming pervasive in today's society and has a presence in self-driving cars, cities of the future, drones, medicine and more. Central to its use and popularity is the combination of Deep Learning and Computer Vision to tackle the important tasks of object classification and localization.

Recent developments of Computer Vision with Deep Learning will be discussed with guidelines & special issues.
overview

- Trends: Vision and Deep Learning architectures
- Running on mobile platforms/embedded devices
- Multi-Modal issues for Vision- a case study
- Temporal Networks - LSTM
- General Adversarial Networks
- Varying data size
- Transfer Learning
- Guidelines
- Resources – datasets, frameworks & computation
- iLab (my) Deep Learning Research Projects

VISION CNN ARCHITECTURES TRENDS
CNN-basics

CNN – has multiple layers that can end with fully connected and softmax layer

- Deeper networks typically can distinguish greater number of objects/classes and/or handle harder discrimination tasks
Shallow CNN for simpler problem of Digit recognition

- 4 convolutional layers + 1 fully connected layer

Going Deeper

ResNet 152 -- has 152 layers !!!!!!!!
Problem – Deeper Network generally slower

- Resnet-152 is 5 times slower than Alexnet

No Fully Connected Layers

There is ALL kinds of research going on in architectures for Deep Learning Networks – sometimes you may have different architectures suited for different problems

- Fully Convolutional Networks for Semantic Segmentation
  (no fully connected layers)

Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmen-
When want to run on device

LOW COMPUTATION – MOBILE AND EMBEDDED DEVICES

MobileNet – uses Depthwise Separable parameters to reduce number of parameters

- Convolution = depthwise + pointwise
  - perform a spatial convolution while keeping the channels separate and then follow with a depthwise convolution.

Why???? Saves on parameters ---LESS to learn, yet still effective
MobileNet –less parameters

• For a depthwise separable convolution on the same example, we traverse the 16 channels with 1 3x3 kernel each, giving us 16 feature maps (called DepthWise Convolution). Now, before merging anything, we traverse these 16 feature maps with 32 1x1 convolutions each (called Pointwise convolution) and only then start to them add together. (this 1x1 is called a depthwise multiplier of 1)

Comparison of number of parameters -- SEPARATED IS LESS!!!

• This results in 656 (16x3x3 + 16x32x1x1) parameters opposed to the 4608 (16x32x3x3) parameters from above.

SO its faster

MobileNet

• You can choose number of layers … example here with 9

After each layer (except FC and softmax) followed by Batch Normalization and ReLU

s2 = stride of 2
Ker°a’s Radar IR PMMW

- Framework = Target mobile and embedded devices
- enables on-device machine learning inference with low latency and a small binary size.
  - optimizing the kernels for mobile apps
  - pre-fused activations
  - quantized kernels that allow smaller and faster (fixed-point math) models.
  - new FlatBuffers-based model file format
- near future will add SSD – localization
- hopefully soon with NN API hardware acceleration
- smaller than 300KB when all supported operators are linked and less than 200KB when using only the operators needed for supporting InceptionV3 and Mobilenet.
- Java and C++ API support

Train using TensorFlow framework then convert

MULTI MODAL (multi sensor)
Multi Modal (multi sensor)

- Have multiple different data from different sensors.
  - CASE 1: Data Aligns spatially
  - CASE 2: Data does not align spatially

CASE 1: data “aligns” spatially

Questions:
- Do you process both in the CNN
- Do you have multiple CNNs? How do you merge results?
- How will this effect Region Proposal (if doing localization)?
- How will this effect speed?

Options:
- OPTION 1: Driven by one modality (e.g. 2D rgb processed first)
- OPTION 2: Process in parallel with separate networks
- OPTION 3: Process together with 1 network
OPTION 1: single data driven $\rightarrow$ use other data later in process


Merge later = Rulebase, SVM, Another FC Network, Algorithmic

OPTION 2: separate CNN run in parallel for each sensor data

Must combine results of separate CNNs:
- Fully connected layer
- SVM
- Rule-based / algorithmic combination
OPTION 1/2 hybrid: rgb 1st for region proposal then both rgb+depth run in parallel

- have RGB + Depth info
- FIRST: Get Region Proposal ONLY from RGB image
- 2 CNNs –
  1 for depth,
  1 for RGB.


Continued...

Option 3: combine data into 1 CNN


4D = r,g,b,depth

Option 3: another example

- Here is a NON visual (but, temporal data) sensor example

WHICH option is best?

- Compared options 1, 2, 3 – for Person Detection & Tracking with RGB + Depth

#1= OPTION 3: 1 CNN for all data
#2= OPTION 2: 2 CNN in parallel
#3= OPTION 1: use only 1 sensor

CASE 2: data does not “align” spatially

- Example speech (not spatial, temporal only) and vision
- Process in parallel with separate CNNs

This works because there is temporal alignment


2014 – Two steam networks


- “Separate” Visual and temporal
- 2 CNNs: 1) for single RGB image  2) for optical flow between current and previous frame(s).
CNN → RNN (Recurrent Neural Networks)

- Used in Speech other naturally temporal data (no spatial components)

RNNs in comparison to a traditional CNN do not capture rich spatial information as well.

- One Idea = Combine RNN and CNN in Sequence

Example from “Show and Tell Me” system


Image Captioning

CNN for Vision process → RNN for language generation

Problems with RNN

- “Long Term” Issue = Difficult to backpropagate an error over a long-range temporal span becomes difficult.

- “Short-term” Issue = Basic RNN does not allow network to “forget” previous hidden states.
SOLUTION ➔ Long Short Term (memory) Networks = LSTM

- Have cells that allow for both “long-term” and “short-term” memory.
- Can propagate without modification using a simple learned gating function and this is a kind of “long range learning”.
- Nodes in an LSTM network allow the network to learn when to “forget” previous hidden states and when to update hidden states given new information. This is a kind of “short-term” memory, basically having an expiration to previous information.

CNN ➔ LSTM

GOAL: Activity Recognition, Image Captioning, Video Description


N × LSTM
OTHER IDEAS—PRE PROCESSING of Temporal Information:
input “temporal segments” → “using 2Stream convolutional system” as input feature vectors into a CNN for recognition


FIRST, extract “temporal segments” used as input to the 2 Network models comparing

• TWO TCLs at each layer each different convolutional filters
• Each layer reduces temporal dimension by half

deep Learning Network
like LSTM

• a 2D matrix composed of feature vectors across different time steps

continued...the Temporal-Inception CNN


• a 2D matrix composed of feature vectors across different time steps
COMPARING: LSTM versus Temporal CNN (inception style)

- Compare Temporal Segment LSTM and Temporal ConvNet as described in C. Ma, M. Chen, Z. Xia, G. AlRegib, "TS_LSTM and Temporal-Inception: Exploiting Spatiotemporal Dynamics for Activity Recognition", 2017
- Performed almost at the SAME level 94.1 versus 93.9 on UCF101 dataset for

GENERATIVE ADVERSARIAL NETWORKS

--- network to teach the network
**GAN: generative adversarial network**

- **2Networks:**
  - 1st network ("discriminator") =
  - takes the image as input and output is the determination of whether the input is a true representation or fake representation of some class.
  - 2nd network ("generator") is trained to generate input to train the first network. The "adversarial" component of this concept is the second network ("generator") tries to progressively create hard input to "fool" the first network.

  - zero-sum or minimax two player game

By doing so, ultimately the first network will achieve better discriminating capabilities, meaning higher accuracy.
GAN –parking example training

**Generator Architecture = Encoder-Decoder Model**


Encoder-Decoder often used for information translation problems, image segmentation, image synthesis

Generator network $G =$ learns to fool the discriminator by generating good “fakes”

**Convolutional Encoder + Decoder**

Idea: decoder = “upconvolutional” layers

GAN –parking example results

Cameras mounted aerially –to be
GAN - applications

APPLICATIONS:
1. may be more suited to future predictive problems such as future video frame predictions
2. discriminator network could be used as a “feature extractor” stage in a more complex CNN.
3. Generator network could be used for synthesis
4. GANs used to produce photorealistic images for modeling scenes, to reconstruct 3D models of objects from images and for texture synthesis [9-11]. They have also been used for various object detection techniques like detection of open spaces for parking [12].


Application:
Resolution Enhancement

--- to resize or modify network

SPECIAL ISSUE OF DIFFERENT DATA SIZE

---
**Option 1 – resize data**

- problem of variable sized input propagates down to the first fully connected/inner product layer which requires a vector of fixed size.

- Resize data to expected input size for Network

**SPECIAL NOTE:** if there are NO Fully connected layers, you do not need to do this. The output layer will be larger but, it does not represent classes but, feature vectors and can potentially be used in same way as before. One example Encoder-Decoder

**Option 2 – “Spatial Pyramid Pooling”**

- “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition” by He et al. proposes a Spatial Pyramid Pooling layer.

- propose to add the Spatial Pyramid Pooling Layer just before the first fully-connected layer (details in the paper).

- hierarchically partitions the feature maps of the last convolutional layer (or the subsequent pooling or response normalization layer) into a fixed number of bins.

- Within these bins, responses are pooled as usually, creating a fixed-sized output
GUIDELINES

Guideline 1

• Decrease size of data slowly

• Bottom $\rightarrow$ Top

  • The spatial resolution $H \times W$ decreases
  • The number of channels $C$ increases
Guideline 2 – Filters

- Receptive field must be large enough to capture objects of interest
- The image region influencing a neuron
- Anything happening outside is invisible to the neuron

- HOW TO INCREASE RECEPTIVE FIELD
  1. Large filters
  2. Chains of small filters

Guideline 3- Filters

- User chain of smaller filters rather than large filter
  - Reduces number of parameters → faster
  - Get same receptive field as the larger filters
  - Get more nonlinearities introduced (example 2 nonlinearities)
Guideline 4- number of filters in layer (= # output channels)

- Be conservative, don’t have too many filters (# filters K below)

Guideline 5 – when computation issue consider separable filters

- Like MobileNet consider separable filters to reduce number of parameters and hence computations.
### Recommendation – transfer learning...

- Don’t have enough data
- Don’t have time/resources to do full training
- Replace final layer(s) and retrain with your own data

### Transfer learning

- Improvement of learning in a **new task** through the **transfer of knowledge** from a **related** task that has already been learned.
- Weight initialization for CNN
## Recommendation Summary

<table>
<thead>
<tr>
<th>ISSUE</th>
<th>RECOMMENDATION</th>
</tr>
</thead>
</table>
| **More Complex Problems**  
- Large number of classes  
- Large data set  
- Greater confusion between classes  |  
- Increase depth of network  
- Current trend – use larger number of layers of small sized filters |
| **Mobile and Low Computational Devices (IoT)**  |  
- For on device computation (otherwise consider cloud)  
- Shallow Networks, minimize number of filters at each convolutional layer.  
- Depthwise Separable Filters (reduce #parameters in mode) – MobileNet architecture [4]  
- Optimize networks (elimination of low contribution nodes, etc.)  
- Consider special purpose frameworks like TensorFlow Lite. [12] |
| **Temporal**  |  
- CNN + Long Term Short Term (LSTM) Networks |
| **Synthesis or Prediction**  |  
- Consider Generative Adversarial Networks |

## Overall Recommendations

<table>
<thead>
<tr>
<th>ISSUE</th>
<th>RECOMMENDATION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Multi-Modal Data</strong></td>
<td></td>
</tr>
</tbody>
</table>
- Tradeoff consideration: potentially higher accuracy for increased computation of combined modality processing versus faster (and potentially lower accuracy) of having separate networks for each modality processing in parallel. |
| **Data Size Variability**  |  
- Pyramid approach [13]  
- When size is different than trained network but, is fixed. Use only front (not FC layers) for feature extraction and then create new FC layers for your new size.  
- Otherwise necessitates rescaling of data to input size of existing |
| **Minimal Time and Resources**  
- Minimal time for training  
- Minimal resources (computation, budget)  |  
- Perform Transfer Learning  
- by finding a pre-trained network that has ideally similarity with your problem and replay the last layer(s)  
- and retrain with your set of classes.  
- Essentially use the pre-trained network’s beginning feature and potentially fully connected layers (minimally replace the end layer) |
## Overall Recommendations

<table>
<thead>
<tr>
<th>ISSUE</th>
<th>RECOMMENDATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of Training Data</td>
<td>• Numerous data sets mostly for 2D rgb images such as ImageNet [23], COCO [24]</td>
</tr>
<tr>
<td></td>
<td>• Emerging data sets in 3D [25-31]</td>
</tr>
<tr>
<td></td>
<td>• Specialized datasets - example: Person datasets: [32-35]</td>
</tr>
<tr>
<td></td>
<td>• Also, may consider use of pre-trained network if you research is focusing on other issues</td>
</tr>
</tbody>
</table>

## Resources

Beyond frameworks
**CNN Frameworks**

- Personal preference: TensorFlow → new TensorFlow lite for mobile devices using hardware acceleration on Android (android neural network api)

This year ...

- **Caffe** (UC Berkeley)
- **Caffe2** (Facebook)
- **CNTK** (Microsoft)
- **Torch** PyTorch (NYU / Facebook)
- **MXNet** (Amazon)
- **Theano** (U Montreal)
- **TensorFlow** (Google)

Newer: Caffe2 and PyTorch, Paddle, CNTK, MXNet

Suggestion: use TensorFlow or PyTorch

---

**Computational Resources**

- Cloud general
- New Google AutoML – As a service (in alpha) - https://cloud.google.com/automl/

- Cloud Services: Google Machine Learning, Microsoft Cognitive Services
  - https://cloud.google.com/ml-engine/

- Intel Movidius Neural Computing Stick
  - has VPU, speed up Ubuntu laptop w/ USB3
  - supports Caffe Framework and uses Intel’s SDK
  - https://movidius.github.io/ncsdk/
Datsets that have segmentation ground truth

- [http://host.robots.ox.ac.uk/pascal/VOC/voc2012/](http://host.robots.ox.ac.uk/pascal/VOC/voc2012/)
- [http://cocodataset.org/#home](http://cocodataset.org/#home)
  - Example from COCO

**Dataset examples**

RGB-D datasets

- Image Net – 14 Million images, 21,841 sub-categories
  [http://image-net.org](http://image-net.org)

<table>
<thead>
<tr>
<th>High level category</th>
<th># subcats</th>
<th>Avg # images per subcat</th>
<th>Total images</th>
</tr>
</thead>
<tbody>
<tr>
<td>person</td>
<td>1,114</td>
<td>1,817</td>
<td>2,003K</td>
</tr>
<tr>
<td>bicycle</td>
<td>1,020</td>
<td>1,817</td>
<td>1,897K</td>
</tr>
<tr>
<td>car</td>
<td>1,990</td>
<td>1,817</td>
<td>3,602K</td>
</tr>
<tr>
<td>bus</td>
<td>1,330</td>
<td>1,817</td>
<td>2,447K</td>
</tr>
<tr>
<td>motorcycle</td>
<td>1,160</td>
<td>1,817</td>
<td>2,149K</td>
</tr>
<tr>
<td>train</td>
<td>1,036</td>
<td>1,817</td>
<td>1,913K</td>
</tr>
<tr>
<td>truck</td>
<td>1,700</td>
<td>1,817</td>
<td>3,148K</td>
</tr>
<tr>
<td>boat</td>
<td>1,098</td>
<td>1,817</td>
<td>2,003K</td>
</tr>
</tbody>
</table>

- COCO – 100 objects, images= 200K labeled, 330K total
  [http://cocodataset.org](http://cocodataset.org)
SPECIAL PURPOSE:
RGB-D People Dataset

- [http://www2.informatik.uni-freiburg.de/~spinello/RGBD-dataset.html](http://www2.informatik.uni-freiburg.de/~spinello/RGBD-dataset.html)
- 3000 images from 3 Vertically mounted Kinect

---

SLR Human attributes dataset

- **Person -rgb +depth (kinect) - 100 people, 100,000 images** [http://srl.informatik.uni-freiburg.de/human_attributes_dataset](http://srl.informatik.uni-freiburg.de/human_attributes_dataset)
ILAB EFFORTS

iSight Goals

• Helping low vision with Mobility by better understanding of their visual world
Seeing Eye Drone

- Replace seeing eye dog
- Computer vision - 3D, deep learning
- Obstacle detection and report

- Using Multiple CNN for User Detection and Heading and Obstacle Detection and Selection for Avoidance
Conclusions

- Follow Guidelines
- “areas to explore”
  - temporal learning
  - Gamming the system - GANs
  - Multi-modal considerations
  - Varying data size
Deep Learning and Computer Vision: Guidelines and Special Issues

Lynne Grewe\textsuperscript{a} and Garrett Stevenson\textsuperscript{a}

\textsuperscript{a}Computer Science, California State University East Bay, 25800 Carlos Bee Boulevard, Hayward, CA USA, 94542

ABSTRACT

The catapult of Computer Vision into recent societal prominence is represented by advancements in self-driving cars, drone autonomy, and cities of the future. Central to these advancements are the developments of Deep Learning with Computer Vision to tackle the important tasks of object classification and localization. This paper surveys some of the current research and presents current guidelines for working in computer vision with deep learning. Additionally, special topics are highlighted including Multi-Modal Vision with Deep Learning and Temporal Networks.

Keywords: Deep Learning, Computer Vision, Multi-Modal Deep Learning, Temporal Networks

1. RESEARCH TRENDS

One Deep Learning research trend is “going deeper”, creating CNNs with a greater number of layers. In [1], VGG Net is a CNN with 19 layers (2014). In 2015, GoogLeNet [2], a 22 layer network, was able to achieve a top 5 error rate of 6.7% and was different than previous CNNs in that is was not a sequentially layered network and instead had parallelly processed layers. At the same time Microsoft ResNet [3] is a 152 layer network that produced an incredibly low error rate of 3.6% for the ILSVRC 2015 challenge. Generally, more complex problems (larger number of classes) can require having deeper networks to yield higher accuracy.

At the same time that some researchers have gone deeper, there is a strong interest in mobile vision and using CNN/deep learning on the mobile devices which are comparatively low computational devices compared to the machines running ResNet. So, going deep (or as deep) will not work on these devices. MobileNet [4] is an example of recent work that creates a CNN architecture that optimizes the network to run more efficiently on mobile devices by using depth wise separable convolution. Following the MobileNet architecture a general guideline for low computational devices is to stick with shallower networks. In [5], use of MobileNet architecture is shown for the iSight system that uses Deep Learning and Visualization to assist people with Low Vision.

Another recent emerging trend is that of Generative Adversarial Networks [6,7]. With GANs there are two networks, the first network (“discriminator”) takes the image as input and output is the determination of whether the input is a true representation or fake representation of some class. At the same time a second network (“generator”) is trained that generates input to train the first network. The “adversarial” component of this concept is the second network (“generator”) tries to progressively create hard input to “fool” the first network. By doing so, ultimately the first network will achieve better discriminating capabilities, meaning higher accuracy. This can be thought of as a zero-sum or minimax two player game. This form of Deep Learning Networks may be more suited to future predictive problems such as future video frame predictions [8] over non-GAN CNNs. Additionally, a GAN discriminator network could be used as a “feature extractor” stage in a more complex CNN. GANs have been used to produce photorealistic images for modeling scenes, to reconstruct 3D models of objects from images and for texture synthesis [9-11]. They have also been used for various object detection techniques like the detection of open spaces for parking [12].

2. MULTI-MODAL VISION AND DEEP LEARNING

Multi-modal vision is the idea that more than one kind of data is being presented to the system. This data may come from multiple sensors and may even be different in nature. Multi-modal data processing with a Deep Learning
framework has not been fully explored but some examples can be found at [13-17]. One technique is to create a CNN only with one sensor data and use the other sensors’ data for additional information. For example, in [17] a system is discussed that performs object detection using a CNN with only 2D image data. Subsequently, two depth sensors (one stereo and the other based on IR technology) have collected depth information of the scene and using the detected location in the 2D image of the object, the 3D location of the object can be estimated. The advantages of such a technique is more simplicity and faster processing than using multi-modal data in the CNN.

In [15, 16], 2D image data and 3D depth information is used directly in the CNN. In [16], an exploration of different ways to incorporate it are compared for performance. Having separate CNNs that run in parallel for 2D and 3D is one case. The second case is presenting 2D and 3D as a 4D image input (r,g,b, depth) into a single CNN. These two multi-modal systems are compared to a 2D only (single sensor data) system. The best results were achieved for the 4D image input. At the same time, this fused data network will be the largest and hence will potentially run slower than the separate CNN case which could potentially run the 2 separate CNNs in parallel. This indicates that presenting all the data at once to a single CNN may yield superior results. Certainly this approach will let the Network learn how to combine the data at different layers (and scales) to best achieve features to yield higher accuracies.

3. TEMPORAL NETWORKS

In addition to looking at different kinds of sensor data, gathering data over time and processing it in a temporal fashion can yield superior results and allow for different applications like activity recognition (“man eating”, “woman walking”) and video captioning (“2 persons walking a dog”). Architectures to incorporate temporal processing in include Recurrent Neural Networks (RNN) and Long Short Term (memory) Networks (LSTM). Recurrent Neural Networks (RNNs) have been successfully applied to processing both speech and video analysis [18] but, are primarily used for speech as they do not inherently capture spatial data as CNNs do. One work looks at integrating CNN and RNN directly in [19]. However, the use of CNNs for temporal processing is best represented in Long Short Term Memory Networks (LSTM) [20-22].

One problem with neural network models using state information is the ability to backpropogate anderror over a long-range temporal span becomes difficult. The LSTM networks have nodes which allow them to propagate without modification using a simple learned gating function and this is a kind of “long term effect” called “long range learning”. Additionally, the nodes in a LSTM network allow the network to learn when to “forget” previous hidden states and when to update hidden states given new information. This is a kind of “short-term” memory, basically having an expiration to previous information. In [20], a CNN is used to perform object identification which is fed into an RNN (textual input) to come up with a image caption. More recently in [22], a CNN is used to process the image frames of a video sequence that is then fed into a LSTM to perform activity recognition, and both image and video captioning.

4. GUIDELINES

Every vision application has different demands. Some systems will have access to good computational resources and others may need to run on mobile and embedded systems. Some systems will have single sensor data and others multi-sensor/ multi-modal data and so on. Table 1 shows some guidelines for different operating scenarios.

<table>
<thead>
<tr>
<th>ISSUE</th>
<th>RECOMMENDATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>More Complex Problems</td>
<td>• Increase depth of network</td>
</tr>
<tr>
<td>• Large number of classes</td>
<td>• Current trend – use larger number of small sized filters</td>
</tr>
<tr>
<td>• Large data set</td>
<td></td>
</tr>
<tr>
<td>• Greater confusion between classes</td>
<td></td>
</tr>
<tr>
<td>Future Looking</td>
<td>• Consider Generative Adversarial Networks</td>
</tr>
<tr>
<td>• Beyond classification of current state, looking to predict future</td>
<td></td>
</tr>
</tbody>
</table>
Mobile and Low Computational Devices (IoT) | For on device computation (otherwise consider cloud)
---|---
- Shallow Networks
- Depthwise Separable Filters (reduce parameters in mode) – MobileNet architecture [4]
- Optimize networks (elimination of low contribution nodes, etc.)
- Consider special purpose frameworks like TensorFlow Lite. [13]

Multi-Modal Data | Tradeoff: potentially higher accuracy for increased computation of combined modality processing versus faster (and potentially lower accuracy) of having separate networks for each modality processing in parallel.

Data Size Variability | Pyramid approach [14]
- When size is different than trained network but, is fixed. Use only front (not FC layers) for feature extraction and then create new FC layers for your new size.
- Otherwise necessitates rescaling of data to input size of existing

Minimal Time and Resources | Perform Transfer Learning - by finding a pre-trained network that has ideally similarity with your problem and replay the last layer(s) and retrain with your set of classes. Essentially use the pre-trained network’s beginning feature and potentially fully connected layers (minimally replace the end layer)
- Minimal time for training
- Minimal resources (computation, budget)

Temporal Processing Needed | Use Long Short Term (memory) Network (LSTM) (or CNN+RNN)
- Video applications, time sequence data

Lack of Training Data | Numerous data sets mostly for 2D rgb images such as ImageNet [23], COCO [24]
- Emerging data sets in 3D [25-31]
- Specialized datasets - example: Person datasets: [32-35]
- Minimal data or none at all.
- Accommodate your sensor data to size
- Scale training data or sensor data to match

Table 1: Current Deep Learning Recommended Guidelines based on Research Trends.

**REFERENCES**

[27] RGB-D SLAM dataset, https://vision.in.tum.de/data/datasets/rgbd-dataset
[29] Big Bird Dataset, http://rll.berkeley.edu/bigbird/
[34] SRL Human Attributes Dataset- http://srl.informatik.uni-freiburg.de/human_attributes_dataset
[35] RGB-D People dataset - http://www2.informatik.uni-freiburg.de/~spinello/RGBD-dataset.html
Outline

- Artificial Intelligence
- Radio Frequency Data
- Big Data
- Research on Big Data
- High Performance Computing (HPC)
- GPU Enabled Target Classification from SAR Imagery
- Summary
Introduction to AI

- **AI** – Machines to think/behave/react - ANN
- **ML** – Data for (Machines) to learn - RL, BN, ILP
- **DL** – Brain-Inspired NN for robust methods – CNN, RNN
  - (mostly supervised from labeled data)

Three Waves of Artificial Intelligence

1st Wave: Handcrafted Knowledge
Humans program systems with explicit rules or logic in limited domains

2nd Wave: Machine Learning
Systems learn statistical models of specific problems using big data

3rd Wave: Contextual Adaptation
Rich collaboration between humans and machines enabled by shared perceptions of the real world

Radio Frequency Data

IBM’s Definition of Big Data

https://www.slideshare.net/EdurekaIN/introduction-to-big-data-hadoop-i
Research On Big Data

- Operational deployment considerations, computation efficiency (SWaP-C)
  - The need for HPC for real-time computing
- Model fidelity complemented with data collections for synthetic-measured data analysis
- Transfer Learning over operating spaces (range, resolution, target settings)
- Big data (volume, velocity, veracity, variety) collaboration policies – what data are accessible for analytics
- Robust evaluation: Validation, Verification, for reproducible results

The Need for Real-time Computing

→ In 90’s, Machine Learning such as Neural Networks was less popular due to various Tech Barriers and Needs
  - Computational Resources were Scarce and Expensive
  - Limited Sensors or Digitized Business Data to be Analyzed

✓ Today, computational resources are not as expensive as in the past; however, abundant of Sensors and Business data needs to be analyzed in Real-time

✓ HPC Enables ML algorithm based decision making in real-time or near real-time
The Advent of HPC

- Since Late 90’s, Computing Technology Has Advanced in an Astounding Pace (The Moore’s Law)

- We are Living in the Age of HPC
  - Faster memory, CPU, I/O communication, and storage as well as compact/smaller size
  - Multi-core Computers
  - Graphics Processing Units
  - Energy-efficient/low-power computing devices (IBM’s TrueNorth)

- More to come
  - Memristor Devices
  - Specialized Chip/cores for Sparse Graph Processing

Recent HPC Hardware Used for ML Algorithms

CPU
Few, fast cores (1 - 16)
Good at sequential processing

GPU
Many, slower cores (thousands)
Originally for graphics
Good at parallel computation

IBM’s TrueNorth

FPGA
Table 1. The number of images of each object at different depression angles.

<table>
<thead>
<tr>
<th>Targets</th>
<th>BMP2</th>
<th>BTR70</th>
<th>T72</th>
<th>BTR60</th>
<th>2S1</th>
<th>BRDM2</th>
<th>D7</th>
<th>T82</th>
<th>ZIL131</th>
<th>ZSU234</th>
</tr>
</thead>
<tbody>
<tr>
<td>17°</td>
<td>233</td>
<td>233</td>
<td>232</td>
<td>256</td>
<td>299</td>
<td>298</td>
<td>299</td>
<td>299</td>
<td>299</td>
<td>299</td>
</tr>
</tbody>
</table>

GPU Enabled Target Classification
Measured SAR Data

- Training, validation, and testing data come from the MSTAR* program sponsored by DARPA and the AFRL in the 1990s
- 10 target classes with images taken at various angles
  - 15 Degree Elevation Angle dataset for training, 17 Degree dataset for testing
  - Roughly 250 images per target class, per angle
  - Generally considered an incredibly small dataset for a deep learning application
- Using a single GPU at AFRL/RI HPC

* MSTAR: Moving and Stationary Target Acquisition and Recognition

Target types

Table 1. The number of images of each object at different depression angles.
SAR Imagery

BMP2  BTR70  T72  BRDM2  BTR60

T62  ZSU234  2S1  D7  ZIL131

Software Tools

• **Python** – Data augmentation methods
• **Caffe** – Deep learning framework employed via DIGITS and command line

---

Proc. of SPIE Vol. 10646  1064601-76

Downloaded From: https://www.spiedigitallibrary.org/conference-proceedings-of-spie on 19 Dec 2019
Terms of Use: https://www.spiedigitallibrary.org/terms-of-use
Caffe

- Deep Learning framework developed by the Berkeley Vision and Learning Center (BVLC)
- Written in highly optimized C++/CUDA code
- Easily define network architectures
- Modify DL models as needed for an application

Caffe ML Algorithm Flow

1. Gather and label data
2. Convert data and labels to LMDB* format
3. Train model in Caffe using training dataset
4. Save learned weights
5. Test model in Caffe using test dataset
6. Evaluate Performance

* LMDB: Lightning Memory-Mapped Database Manager
Clean training run

Neural Net reaches over **99% accuracy** on validation set

Classification results on Measured Data

**~99% accuracy** on 10-target classification using Caffe

State-of-the-art results

Key network parameters

Learning rate 0.001
Batch size 64
1000 training epochs

5 Convolution layers
3 InnerProduct (FC) layers
2x2 stride 1 max pool filters

Dropout regularization
Target Classification Using DNN on Synthetic SAR Data

- Training, validation, and testing data used from Synthetic Radar Data
- **30 target classes** with images taken at various elevation angles and a single azimuth angle
- Instead of Backprojection Image formation, we used Range-Doppler Map of the Targets
- **We found about 99% accuracy on Target classification**

Target Classification Using DNN on Synthetic and Measured SAR Data

- The objective of this research is to evaluate performance of target classification using Synthetic vs. Measured SAR data (or vice versa) and identifying the “Gap/Tech Challenges” to generate High Fidelity Synthetic SAR data
- We implemented Training on measured SAR data for three targets and Tested on Synthetic SAR data (of the same targets)
- **We found very low accuracy on Target classification**
- This is due to the fact that quality (i.e. NIIRS) of synthetic data must be very close to measured data
  - This will require huge HPC resources and expertise in Computational Electro-magnetic

➢ TRANSFER LEARNING
Summary

• Research on applying DL techniques to multi-sensor information fusion is evolving

• Followings are key research that needs to be addressed:
  – Filling the Gap/mismatch between measured and synthetic data
  – Transfer Learning over operating spaces (range, resolution, target settings)
  – Robust evaluation of the algorithms
  – Operational deployment considerations, computation efficiency (SWaP-C)

References

Deep Learning for Object Recognition from High Volume Radio Frequency Data

Uttam K. Majumder\textsuperscript{a,1}, Erik P. Blasch\textsuperscript{b}

\textsuperscript{a}Air Force Research Laboratory, Information Directorate, 525 Brooks Road, Rome, NY13441.  
\textsuperscript{b}Air Force Office of Scientific Research, Wright-Patterson AFB, OH 45433

ABSTRACT

Much research efforts have been devoted to applying deep learning (DL) algorithms in video imagery for object recognition. However, very limited publications can be found on technical challenges and approaches to execute DL algorithms in radio frequency (RF) data. This talks highlights recent advancements of DL on synthetic aperture radar (SAR) imagery for object recognition. Radar enables imaging ground objects at far greater standoff distances than an electro-optic sensor. Further, radar enables imaging a scene and obtaining salient features of objects in all weather/day-night conditions. One example is that future self-driving/autonomous vehicles/cars could integrate radar among other sensors for decision making while sharing the roads and avoiding collisions/accidents. Existing non-DL based object recognition algorithms are less accurate and require impractically large computing resources. DL enables more accurate, real-time/non-real-time, and low-power object recognition system development. An examples is presented on Convolution Neural Network (CNN) based SAR object recognition for GPU and energy efficient computing systems. Results demonstrate acceptable classification accuracy on relevant SAR data.

Keywords: Deep Neural Network (DNN), Artificial Intelligence (AI), Synthetic Aperture Radar (SAR), Radio Frequency (RF), Big Data

RF BIG DATA

According to IBM\cite{IBM1, IBM2}, big data has several characteristics. Among these are volume, velocity, variety, and veracity. A big volume of data could be processing terabytes to exabytes or more data in a milliseconds or seconds. In some applications, RF data could be collected from petabytes to exabytes and these data need to be processed (e.g., object recognition) in real-time or near real-time. Radar images are difficult for humans to analyze. It requires significant efforts for accurate interpretation; however a DL system is very capable to interpret these images. The benefits of utilizing radar technology are evident given its advantages over electro-optical imaging. Specifically, radar is able to operate in a variety of operating conditions—including poor visibility, inclement weather, and night-time settings. Given the disparity between the technical capabilities of radar and its difficulty of interpretation, it is a goal to develop accurate methods for automatically recognizing objects in radar images without the need for expensive expert analysis. The benefits of object recognition for radar imagery include developing self-driving cars to autonomous systems.

\textsuperscript{1} Correspondence: ukmccny@gmail.com
Though most recent work in applying artificial neural networks (ANN) to computer vision has focused on electro-optical images, the application to radar imagery is of particular interest for our research. Previously, achieving RF object classification using DL methods was the absence of necessary computational power. A radar imagery data collection project—called Moving and Stationary Target Acquisition and Recognition (MSTAR) program—was funded by DARPA and the AFRL in the 1990s to fill the void in available radar data for object classification research[3, 4]. Additionally, the recent introduction of affordable GPU computing resources[5] has made the efficient processing of datasets for deep learning (DL) applications a reality for RF object classification research efforts.

**CONVOLUTIONAL NEURAL NETWORKS**

Convolutional neural networks (CNN) are a special case of neural networks, as they make the explicit assumption that the input data are of constant size. This enables certain features to be encoded in them. In particular, the computational units, or “neurons,” share parameters with their neighbors. The connections conceptually result in an intuitive visualization of the network as a series of filters that “slide” or “pool” over regions they are connected to—producing mathematical output based on the parameters, or “weights,” of each filter. The output produced by these filters is then fed to other filters connected to them, and so on, concluding with a fully connected dense network shown in Figure 1.

![Figure 1: Illustration of Convolution Neural Networks](image)

Through training, these filters learn to respond to complex features in image data. Filters closer to the data input layer learn to recognize simple features, such as edges. Mid-level filters might learn compositions of edges, like simple shapes. High-level filters then learn complex abstractions of the data, such as compositions of shapes forming full objects of interest. Thus, CNNs are a natural fit for working with the conceptual hierarchy of features present within image data.

**EXPERIMENTS ON RF OBJECT CLASSIFICATIONS**

Software tools used for this research are: Caffe, DIGITS, Python, and LMDB. Caffe is a deep-learning framework maintained by the Berkeley Vision and Learning Center (BVLC)[7]. Caffe provides full end-to-end capability for deep learning tasks, and is the deep learning framework of choice for many in the machine learning community.

Experimental data were extracted from the MSTAR public dataset [3]. The MSTAR public dataset contains several hundred SAR images of a variety of targets collected in varying conditions. Sandia
National Laboratory used an X-band radar sensor at 1-foot resolution in spotlight mode to collect target data at 15, 17, 30, and 45-degree depression angles. Images taken at varying azimuth angles provide a 360-degree view of each given target. The images are 128x128 pixel chips containing a target roughly centered within its background. Bulldozers, trucks, tanks, and anti-aircraft vehicles are among the targets included in the dataset as shown in Figure 2 with the SAR images in Figure 3.

![Figure 2: Electro-optic Images of Objects](image1)

![Figure 3: SAR Images of Objects](image2)

**RESULTS**

On MSTAR public release data [3], 99% classification accuracy was achieved using the model specifications described in Chen, *et al.* [8]. Most DL architectures tested reached an accuracy limit of 98.3%-98.7%, depending on testing batch size and number of testing iterations ran. This held true for models employing small convolution filters, models with additional convolution layers before pooling, and models with fully connected layers at the end of the network architecture. Dropout regularization provided an accuracy improvement of roughly 0.5% over models not employing a DL technique. Though a relatively modest improvement, it is nonetheless indicative of the benefits of using dropout to ensure a model’s ability to generalize to new data.
CONCLUSION AND FUTURE RESEARCH

Research on applying DL techniques to multi-sensor information fusion is evolving, such as for image fusion [9]. Integrating RF sensor for Artificial Intelligence (AI) and information fusion has lots of technical challenges. Followings are key research that needs to be addressed:

- **RF Synthetic Data Research**: DL algorithms require lots of training data. Hence, research needs to be conducted to develop high fidelity synthetic RF data to augment measured RF data. This is important as measured RF data are expensive to collect.

- **Transfer Learning Algorithms Development**: Develop transfer learning techniques over operating spaces (e.g., range, resolution, target settings).

- **Robust Evaluation of DL Algorithms**: Results (classification accuracy) of DL algorithms are meaningless unless they are validated with representative operating environments (e.g., environment) [10]. Hence, validation and verification for reproducible object classification results in extensive operating conditions are very important.

- **Computational Efficiency**: Developing a real-time training algorithms and size, weight, and power-constrained (SWaP) computing systems will be required for future DL-based AI systems.

REFERENCES


DISCLAIMER

*The views expressed in this article are those of the author and do not reflect official policy of the United States Air Force, Department of Defense or the U.S. Government.*
Retrospectives on the Applications AI and Deep Learning in Information Fusion

Ivan Kadar
Interlink Systems Sciences, Inc.
1979 Marcus Avenue, Lake Success, NY 11042

BACKGROUND-PERSPECTIVES

In this position paper concrete examples of the use of Neural Networks (NNs) and Artificial Intelligence (AI) components in Information Fusion (IF), are presented, based on work performed in the 1980s and 1990s-on at Grumman Aerospace Corporation, Advanced Technology Development Department, in Bethpage NY.

While the application of AI was prominent before NNs became popular, this paper starts with a description of my short verbal introduction to the subject at the panel, viz., NNs; and subsequently expands on the subject to depict the application of AI and NNs for IF applications.

As further detailed below, as part of the work at the Advanced Technology Development Department, of the Knowledge-Based Processing Systems Group and associated Knowledge-Based AI Technology development laboratory (1984-1990), founded and managed by the author starting in 1983, we used: tracking, association and fusion algorithms, and developed: e.g., Common-LISP-based algorithms/programs/systems, including an interactive digital LISP-based image processing facility for both low-level vision processing and for an initial capability of high-level image understanding towards target recognition; used Expert Systems Tools [1], NNs [2], Dempster-Shafer theory (DST)-based algorithms for evidence representation and combination/fusion [3] and Fuzzy-Set [4]; the latter three were also considered part of AI and not IF during that time frame. Given the available tool sets, we also developed the architecture, designed and implemented the simulation of a Tactical Assessment Expert System (TAES) [5]; all delineated in the sections below.

1. NEURAL NETWORKS

During my short introduction at the panel discussion the subject being addressed and the participants, only highlighted one aspect of the subject: “Deep Learning” [6, 7], viz., the use of Neural Networks (NNs) for significant feature(s) recognition back in the 1987-1990’s-on.

During that time frame we used a three hidden layer abductive polynomial NN [8] as the feature selection component of a ground-based-emitter discrimination algorithm in mid 1990s, resulting in a US Patent in 1999 [9]. Other components of the algorithm included Fuzzy Sets [4], and a related classifier.

An example abductive polynomial NN is shown below [8]:

![Diagram of Neural Network]

where, in the equations shown in the algebraic forms below, the w’s are the weights learned by regression, and the x’s are the input variables.
Singles, doubles, and triples are elements whose names are based on the number of input variables. The algebraic form of each of these elements is shown in the following equations:

\[
\text{Single} = W_0 + (W_1 X_1) + (W_2 X_1^2) + (W_3 X_3)
\]

\[
\text{Double} = W_0 + (W_1 X_1) + (W_2 X_2) + (W_3 X_1^2) + (W_4 X_2^2) + (W_5 X_1 X_2) + (W_6 X_1^3)
\]
\[
+ (W_7 X_1^2 X_2)
\]

\[
\text{Triple} = W_0 + (W_1 X_1) + (W_2 X_2) + (W_3 X_3) + (W_4 X_1^2) + (W_5 X_2^2) + (W_6 X_3) + (W_7 X_1 X_2)
\]
\[
+ (W_8 X_1 X_3) + (W_9 X_2 X_3) + (W_{10} X_1 X_2 X_3) + (W_{11} X_1^3) + (W_{12} X_2^3) + (W_{13} X_3^3)
\]

Given the CPU clock speeds of the mid 1990s, the training alone took two days. That is, computers were very slow (100-300 MHz clock speeds), memory was scarce and very expensive, and even the three hidden layers network took a day to converge to the solution.

The term “Deep Learning” [6, 7] (referring to hidden layers in NNs) was not introduced at that time although the network had three hidden layers, and indeed it “deep” learned.

Of course, today’s Deep Learning NNs [7, 10] have multitudes of hidden layers for features recognition, but even the fastest special-chip-based networks do not always converge in real-time use quickly enough, (as mentioned in articles published in the Technology Related Business Sections in the New York Times during the Spring 2018).

2. ARTIFICIAL INTELLIGENCE (INCLUDES NNs)

Sections below, illustrate the complementary interactions and harmonious use of AI and NNs components in IF applications. Part of the material is adapted in part from the author’s position paper within: “Results from Levels 2/3 Fusion Implementations: Issues, Challenges, Retrospectives and Perspectives for the Future – An Annotated Perspective” presented at the SPIE Signal Processing, Sensor Fusion and Target Recognition XVII conference, Proc. SPIE Vol. 6968, Orlando Fl., April 2008.

It is well known that the concept of Situation Awareness (SA), (Level 2), and Threat Assessment (TA), (Level 3); SA/TA existed before the Joint Director of Laboratories Fusion Model (JDL) [11, 12, 13], viz., JDL established the numerical representation and “at-that-time” definition of Fusion “Levels” [9]. This is illustrated in a “circa 1984-1986 vintage” simulated “Tactical Assessment Expert System architecture,” [14] depicted in Figure 2.

The TAES system utilized knowledge-based Expert Systems tools combined with numerical algorithms with the primary objective to reduce pilot workload so the pilot becomes the systems manager not just the operator.

Figure 2. Tactical Assessment Expert System Functional Architecture
That is, the TEAS system could inform the pilot: targets detections/IDs, environmental data and system status reports, and suggested tactical decisions for pilot’s override. This was accomplished by modeling and encoding pilot’s thought processes (via interviews) to form of an “experienced-software-copilot” during the SA/TA knowledge acquisition stage. The fundamental construct/design guidance for this system was based on an early fusion model (before JDL), called the “Perceptual Reasoning Machine (PRM) paradigm” [15], shown as an internal governing component of the Generic Information Fusion Process Model System (PMS) [14,15,16,17] shown in Figure 3. The dotted and yellow-highlighted blocks shown in Figure 2, correspond to the PRM functions shown in Figure 3. Figure 4 depicts the information flow among the PRM elements.

Figure 3. Genetic Information Fusion Process Model System (PMS)

Figure 4. Information Flow among PRM Elements [Note: KB=Knowledge Base]
The PRM construct, depicted in Figures 3 and 4, can be viewed as a “meta-level information management system”, in general, and specifically when used within PMS, which is, a set of procedures and algorithms that capture the functional (temporal and/or spatial) dependency relationships of the task or processes being modeled [14-17]. PRM consists of a feedback planning-resource control system whose interacting elements are: “gather/assess”, “anticipate” and “predict” [15-17], which are detailed, along with the required knowledge-bases, in Figure 4. Note: the “gather” part of the “gather/assess” module (shown in Figures 3 and 4) fuses optimally weighted information from multiple sensors/sources, and “assess” part functions are depicted in Figure 4.

2.1 Evolving tools for Levels 2/3

The TEAS software architecture, shown in Figure 2, was built on Common LISP and using the Automated Reasoning Tool (ART) [18] an expert system building tool. There were several knowledge bases (KBs) as shown in Figure 2, (1) Declarative KB (functional - not shown) consisting of: Static KB- relations via semantic network using inheritance wherein the system exploits the inheritance structure of the KB to interpret incomplete data and a Pop-up menu driven on-Line KB to keep track of targets encountered, and to support interpretation of incomplete data; and, (2) Procedural/Operational KB consisting of: Production rules, interacting with a Dynamic KB using mathematical constructs - associated with the Control KB which are coupled with the interacting feedback structure of the PRM components of: Dempster-Shafer evidential reasoning (part of ID fusion expert system), Data Fusion/Tracking and Anticipation Expert Systems providing the reasoning mechanism for TEAS to arrive at a comprehensive interpretation of uncertain situations. The TEAS system was totally data Driven, i.e., LIFO - rules groups fired independently based on available data, allowing all modules to access to information during any stage of the program. Simulation results illustrated the interaction between a hypothetical scenario pilot thought process model (database derived from Jane’s Book in all the Worlds Aircrafts) and the system, using simulated sensor reports to handle uncertainty. The TEAS system ran on the Symbolics 3675 LISP machine. The question arises how would one implement the software architecture of TAES today and what has changed since it was built.

As evident from the TAES construct, early approaches to higher-level fusion evolved from the mainstream use of early expert system tools (e.g., Knowledge Engineering Environment, “KEE” [19]. Automatic Reasoning Tool, “ART” [18] built in Common-Lisp, both rule-based providing forward and backwards chaining, while ART provided hypotheses generation capability and prediction), other tools were based on: strings oriented symbolic (objects-oriented) language (SNOBOL-4) for pattern matching, common-LISP, logic (PROLOG), logical templates, procedural-LISP-based, such as Procedural Reasoning System “PRS”, case-based languages, Blackboard (BB) system [20] representations, associative memory [21], schema-based languages and neural networks (NN) for knowledge elicit/learning/acquisition (viz., background NNs learning the pilot’s functions), evidential reasoning and ID declarations fusion using Dempster-Shafer calculi, tracking and related association algorithms, along with some of the basic methodologies remaining a part of current approaches. It should be noted: the “anticipate/predict” module of PRM (see Figure 4) was initially implemented using a KB of prior domain knowledge (which is automatically updated with current/latest knowledge), an inference engine and ART. Subsequently it was modified and used an associative memory [21] NN provided at that time by DEC corporation. The associative memory provided the “perceptual reasoning associative recall” function [22] in the PRM.

Current, and potential future trends, are primarily based on agent-based models [23] of interactions, including Blackboard (BB) systems [20], NN behavioral learning systems for knowledge acquisition, ontology representations (extending schemas), probability (Bayes-nets and Dempster-Shafer calculi and its extension [24, 25, 26] and possibility (fuzzy-sets)-based methods [27], graph theory oriented relational representations, game theoretic methods of optimization, some coupled with influence diagram formulations [28], but not excluding rule-based expert system tools, such as CLIPS built using C and JAVA [29], with the above representing a non-exhaustive representative list. The author is not aware of any comprehensive studies to compare the efficacy of the “historical main stream” and “current-main stream” trends in order to learn from experience.

2.2 Knowledge representation and reasoning (KRR) approaches/issues, when AI and NNs are available

Related to section 2.1, methods of knowledge elicitation/acquisition, learning, representation and reasoning (KRR) have not appeared to have made significant strides over the past several years in spite of several conferences devoted to KRR [30], illustrating the difficulty associated with this topical area. The following list highlights potential KRR issues and challenges [30, 31]:

Ixxviii
• Adequacy of KRR
  – Using logic, semantics, ontology, probabilistic methods, neural networks, associative memory, blackboard, simulations, rules and computation - how to quantify and measure?

• Expressiveness of models vs. tractability of inference
  – Measures of richness of model vs. knowledge that inference is decidable and will produce and answer efficiently; and why correct and how arrived at that answer? [31]

• Managing Complexity
  – Limits about tractability - how to bound the problem with incomplete knowledge

• Data Information
  – How to manage heterogeneous and uncertain Knowledge Sources, and detect duplicate or incomplete concepts
  – Knowledge Acquisition/Elicitation Issues
    • Expert’s difficulty in verbalizing knowledge
    • Reliability and uncertainty of knowledge, and how to calibrate (ground truth)
    – Methods for reasoning and discovery under uncertainty
    • Indirect learning of knowledge - “on-line background” learning of “selected” features

• Presentation of knowledge to different users/experts with different levels of expertise. i.e., what is pragmatic?

3. CONCLUSIONS

The purpose of this position paper is to illustrate the 1987-1990’s use of NNs “Deep Learning” and AI algorithms, and subsequently highlight the 1984-1987-on retrospectives and perspectives on issues and challenges of Levels 2/3 information fusion using AI and NNs methods as components of the implementations, by presenting an independent point of view. There are many other possible additional implementation issues and challenges remaining, for example, in: model refinement, computational and processing methods, optimization, automation and decision making under uncertainty, human-machine interface and integration, distributed systems, knowledge elicitation, deep learning and representation, and potentially many more issues and challenges that hopefully will be addressed as part of future research in this area using new approaches.

REFERENCES

[26] Dezert-Smarandache Theory of Plausible, Paradoxist, and Neutrosophic Reasoning.mht
[29] www.ghg.net/clips/CLIPS.html A Tool for Building Expert Systems