TOWARD AUTOMATED AERIAL REFUELING: AUTOMATED VISUAL AIRCRAFT IDENTIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS

THESIS

Robert L. Mash, Captain, USAF

AFIT-ENG-MS-17-M-048

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-17-M-048

TOWARD AUTOMATED AERIAL REFUELING: AUTOMATED VISUAL AIRCRAFT IDENTIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Robert L. Mash, B.S.E.E.

Captain, USAF

March 2017

AFIT-ENG-MS-17-M-048

TOWARD AUTOMATED AERIAL REFUELING: AUTOMATED VISUAL AIRCRAFT IDENTIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS

THESIS

Robert L. Mash, B.S.E.E.
Captain, USAF

Committee Membership:

Lt. Col. John M. Pecarina, PhD
Chair

Dr. Brett J. Borghetti
Member

Maj. Brian G. Woolley, PhD
Member

Dr. Scott L. Nykl
Member

AFIT-ENG-MS-17-M-048

Abstract

In the military domain of autonomous aerial refueling operations, automated visual recognition of an approaching aircraft critically supports mission goals. This work leverages recent developments in the field of pattern recognition in natural images with deep convolutional neural networks (CNNs). The first article reviews the operational details of CNNs, then demonstrates a hyper-parameter optimization process. The second article investigates advanced forms of data augmentation in terms of image recognition performance. Finally, the third article demonstrates a novel ensemble confidence measure as well as a modified ensemble compression technique which retains a useful confidence measure in a single *student* network.

To my lovely Dorothea. Thanks for your patience and support.

Acknowledgements

I would like to express my sincere appreciation to Lt Col John Pecarina, Maj Brian Woolley, and Dr. Brett Borghetti. Thank you for your guidance and professional support, without which this research, and my personal journey into the field of deep learning, would not have occurred.

This work was supported by the Air Force Research Lab's Sensors Directorate, Layered Sensing Exploitation Division and the Aerospace Systems Directorate, Power and Control Division.

Robert L. Mash

Table of Contents

Page

Page

List of Figures

List of Tables

TOWARD AUTOMATED AERIAL REFUELING: AUTOMATED VISUAL AIRCRAFT IDENTIFICATION WITH CONVOLUTIONAL NEURAL NETWORKS

I Introduction

Background

In order to maintain air dominance in a changing global and technological landscape, all the while doing more with less, the DoD envisions an Air Force with increasing numbers of autonomous aircraft. Distinct from remotely piloted aircraft, an autonomous aircraft is expected to plan, manage, and adapt its own activities in collaboration with a human decision maker working to accomplish mission objectives.

Autonomous aircraft are consistent with the DoD's Third Offset strategy leveraging innovation in the fields of machine intelligence, human-machine teaming, and network-enabled operations as in [1]. These aircraft are expected to expand projection of air power while reducing risk to human aircrew. Further, if reductions in total personnel per aircraft can be realized, higher operational tempos may be possible while avoiding additional long term support costs.

Adoption of the Third Offset strategy by USAF leadership implies a willingness in the future to accept truly unmanned aircraft operating in a collaborative, flexible, human-machine Command-and-Control (C2) structure. This notion of trusting a machine intelligence to when appropriate, take on an increasing role in the Observe-Orient-Decide-Act (OODA) loop decision process represents a revolution of sorts both in terms of policy and technology. This thesis addresses specific technologies that play

a pivotal role in such autonomous systems.

Any realization of this Third Offset inspired vision of autonomous aircraft operations hinges on continued development in several core technology areas. Specifically, advancements in the field of machine intelligence will be required to realize these capabilities. An autonomous aircraft piloted by a machine intelligence must have and use a robust situational understanding in order to accomplish mission objectives.

Situational understanding, in turn, requires accurate and automatic pattern recognition in order to transform data into knowledge, thus maintaining a firm grasp of the surroundings of the aircraft.

Of course, a machine intelligence appropriate to pilot an aircraft autonomously will require many other advances in artificial intelligence beyond improvements in pattern recognition. The present research contributes only to a small part of the progress represented by a machine intelligence on the scale required for a truly autonomous aircraft. The objective of this work is to apply recent academic advances in visual pattern recognition with deep, artificial neural networks to the challenge area of improving autonomous situational understanding.

To be clear, since approximately 2012 there has been an upheaval of sorts within the field of computer vision with artificial neural networks, and in particular, convolutional neural networks.

Key developments include improvements in ANN training algorithms and methodologies such as dropout as in [2], convolutional neural network specific structural enhancements like max pooling as in [3], inexpensive access to high performance computation with graphics processing units, and generally increased availability of large training datasets on the open internet.

As with many computationally intensive fields of study, the declining cost of high performance computation has been key to the development of modern artificial neural

networks, of which convolutional neural networks are a class. Particularly, modern general purpose graphics processing units are crucial for training state of the art convolutional neural networks.

Particularly, the areas of object detection, classification, and segmentation have all undergone a step-change in performance approaching, and in some cases exceeding that of humans. Consequently, these advances have been embraced and improved upon by the private sector in collaboration with academia. Such recent developments motivate an exploration of how artificial neural networks might be applied to the pattern recognition challenges of fielding autonomous aircraft.

Problem Statement

This research is situated in the domain of autonomous aerial refueling. In this domain, a future autonomous refueling tanker must find, fix, and track other aircraft in the lead-up to precision formation flight while performing aerial refueling. Automation of such a scenario requires that many activities presently entrusted to trained aircrew instead be performed by algorithms.

During aerial refueling, an aircraft to be refueled approaches a tanker aircraft from below and behind in a predefined flight path. A refueling tanker equipped with an automatic aerial refueling vision system would have rear-facing cameras aligned with the standard approach vector. This would result in a perspective of the approaching aircraft on refueling approach as viewed from the front and above, or the refueling perspective. This refueling approach imagery is input to a system which generates an evaluation of the aircraft type for each video frame, which drives the operation of the other autonomous systems in the refueling process.

The present research attempts to apply convolutional neural networks, a subset of the general field of artificial neural networks, to the problem of robust visual aircraft

identification. The over-arching investigative question is this: Can the current state of the art in object classification identify aircraft on approach as reliably as a member of trained aircrew?

Research Objectives

It is hypothesized that a convolutional network, sufficiently and properly trained, will succeed in this task in much the same way that a member of an aircrew learns to identify aircraft visually given enough prior examples. Investigation of this over-arching research question is broken up into three sequentially related areas of inquiry: optimal hyperparameter selection, optimal data augmentation, and ensemble combination and construction.

Optimal Hyperparameter Selection.

Convolutional neural networks represent the current state of the art in object classification, with results on par with human performance as analyzed in [4] and [5]. However, best engineering practices in this burgeoning field remain under-defined. Questions abound as to how to optimally design and train a network for a specific classification domain. Consequently, a first investigative sub-question is this: what set of network design hyper-parameters, which describe the network shape and capacity, optimize object classification performance in the specific domain of aircraft recognition? It is hypothesized that general principles from the field of machine learning, of which deep learning is a sub-field, can be applied to arrive at an optimal network configuration.

Optimal Data Augmentation Method.

In the field of deep learning, data augmentation is a technique that is used to improve the ability of a neural network to usefully generalize over a broader range of possible inputs than is represented by the training dataset. As examined in [6], each exemplar training image represents a particular location on a high dimensional latent function which is learned by the network during training. In order to improve the generalizing capability of the network, label preserving transforms are randomly applied to each exemplar training image in order to represent locations on the latent function neighboring that of the original image. A second sub-question explored is this: What combination of data augmentation techniques is most applicable to the domain specific task of classifying aircraft? It is hypothesized that randomly applied rotations, occlusions, and scaling will contribute positively to test performance. These are however, computationally complex transformations. Therefore, improvements in performance must be pragmatically balanced against prohibitive computational complexity.

Ensemble Combination and Compression.

Ensembles of trained neural networks reliably yield significantly better test performance than individual networks as demonstrated by [7] and [8]. Also, the plural structure of ensembles naturally provide a robust plurality-vote based ensemble confidence value which is correlated with the ensemble's output correctness. As examined in [**?**] and [9], ensemble confidence is useful for implementing a deferring-action decision framework when classification confidence is low. Unfortunately, the performance improvement and robust confidence of an ensemble comes at the cost of increased computational complexity in proportion to ensemble size. A third research sub-question is this: Can existing techniques in ensemble combination and compression be modified

to yield a computationally manageable network while preserving both the favorable performance of a large ensemble as well as a robust confidence measure? It is hypothesized that the reliable ensemble combination techniques leveraged throughout the image processing community will yield substantial improvements over any single network trained to recognize aircraft. It is further hypothesized that the image-classifying and confidence generating functions encapsulated in the ensemble can be usefully approximated in a single student network that later proves advantageous in the inevitable Size-Weight-Power-Cost aircraft integration trade-off process.

Expected Contributions.

Expected contributions in this thesis are several. First, from the optimal hyper-parameter selection experiment, a proven set of neural network configuration hyper-parameters are expected to result from application of machine learning fundamentals. Also, in order to train and evaluate such a neural network, this experiment is expected to produce an appropriate aircraft recognition dataset.

Next, the optimal data augmentation experiment is expected to produce a list of training image augmentation techniques ordered by the engineering advantage conferred in terms of test set accuracy. These data augmentation techniques are applicable to the domain of aircraft recognition and may be applicable in a broader superset of domains.

Lastly, from the third experiment involving the combination, and compression-with-confidence of an ensemble of convolutional neural networks, a *student* network is expected yielding most of the ensemble's classification performance. Further, the student network is expected to yield a robust estimate of the *teacher* ensemble's plurality-vote based confidence, in support of a reject-option[10] based decision framework.

Thesis Overview

Under the auspices of AFIT's Automation and Navigation Technology center, the present research has been conducted in support of the autonomous aerial refueling project in the area of convolutional neural networks.

Experimental Methodology.

The experiments described in this thesis are primarily implemented with the PyCaffe Python interface to Caffe. Additionally, various open source libraries to include Sci-Kit Image, matplotLib, and others were heavily used. Caffe, as introduced in [11], is a product of the U. C. Berkeley Vision and Learning Center.

In the first experiment, an imagery dataset roughly evenly representing ten USAF aircraft was sourced from Flickr.com groups and Youtube.com videos. This 76,000 image dataset is limited to day-time, out-door photography and is taken from a broad, but not all-inclusive range of perspectives. Specifically, the refueling perspective, or that perspective from the refueling boom operator on a leading refueling aircraft, is under-represented. This imagery dataset was developed in the absence of any existing USAF refueling perspective datasets.

Variants of the popular Alexnet convolutional neural network from [12] were formed by retaining the structure of the convolutional layers of Alexnet, and adjusting the hyperparameters associated with the fully-connected classifier portion. These networks, with varying capacity, were individually trained using the learning techniques cited in [12]. Of these variants, the one with the highest test accuracy (91.5%) was selected and named AfCaffe. This network structure is subsequently used in the remainder of the present research. AfCaffe was implemented with the open source Caffe Deep Learning Framework from the U. C. Berkeley Vision and Learning Center.

As a consequence of training with the all-perspective dataset, the instances of AfCaffe have only a limited ability to recognize aircraft from the refueling perspective. In the future, a dedicated refueling perspective dataset should be developed by partnering with the USAF test community centered around Edwards AFB and Eglin AFB. A convolutional neural network could then be trained on that dataset with the techniques proven in the present research, presumably yielding a network with favorable performance solely from the tanker aircraft refueling perspective.

In the second experiment, combinations of random training data augmentation techniques including random scaling, rotations and occlusions are evaluated using cross validation techniques from the field of machine learning. During the training process, the proposed techniques randomly transform each training image prior to presentation to the neural network. The effect is to improve the generalizing capability of the network by effectively multiplying the size of the training dataset. Several different instances of AfCaffe were trained and their test performance comparatively evaluated. Random image scaling with randomly drawn occlusions yielded 93.5% test accuracy, or a 45.7% reduction in test error over the simple methods cited in [12].

In the third experiment, the several differently trained, but structurally identical instances of AfCaffe from work in the first two experiments were formed into a neural network ensemble. Next, the latent representation of the knowledge stored in the ensemble was compressed into a single instance of the common Alexnet CNN using the technique explored in [13]. The compression algorithm effectively mitigates the prohibitive computational complexity of the ensemble while preserving the ensemble's substantial performance improvement over any single ensemble member. Further, Caruana's compression algorithm was modified to teach the student a function that approximates not only the primary classification task encoded in the ensemble, but also an estimate of the ensemble's plurality-vote confidence score.

Assumptions / Limitations.

The all angles dataset is sourced from general populace photography, and naturally has representation gaps. It is therefore not well suited for training a convolutional neural network to consistently perform well from the perspective of a refueling tanker. However, the techniques in network sizing with hyper-parameters, data augmentation, ensemble combination, and network distillation are still fully valid. Therefore, any subsequent research effort utilizing a different dataset will have concomitant strengths and weaknesses related to the latent representation in that dataset.

Thesis Structure.

This thesis follows the scholarly article format containing three separate works following in chapters two, three and four.

Chapter two is reproduced from proceedings of the 2016 IEEE National Aerospace and Electronics Conference (NEACON), "Toward Aircraft Recognition with Convolutional Neural Networks" which led development of a visual aircraft classifier based on Alex Krizhevskys famous Convolutional Neural Network (CNN). Contributions from this paper include developmeSo nt of an internet sourced dataset of 76k images of 10 USAF aircraft, and AfCaffe, a bias-variance optimized variant of Alexnet from [12] which yields better aircraft recognition test performance than [12] with a significant reduction in time and space complexity.

Chapter three is reproduced from Proceedings of the 12th International Symposium on Visual Computing, "Improved Aircraft Recognition for Aerial Refueling through Data Augmentation in Convolutional Neural Networks". This work utilized a novel combination of random scaling and rotation [14] data augmentation techniques to demonstrate an additional 45% reduction in test error over the standard randomly applied horizontal flipping and randomly located fixed-size cropping data

augmentation methodology used in training AfCaffe.

Chapter four reflects a journal article tying together the research areas of ensemble-compression and confidence-based-decision-frameworks. First, an ensemble is formed from the individually trained instances of AfCaffe from chapter III. Next, ensemble compression is accomplished using a modified version of the techniques developed in [13]. The effect is to approximate the classification performance of the ensemble, which increases its competitiveness in terms of the Size-Weight-Power-Cost aircraft integration trade-off. Also, the student network learns to approximate a robust, plurality-vote derived ensemble confidence measure, useful for implementing a high reliability decision framework as explored in [10] by Chow.

II Scholarly Article: Toward Aircraft Recognition with Convolutional Neural Networks

Robert Mash, Nicholas Becherer, Brian Woolley PhD, John Pecarina PhD

Proceedings of the 2016 IEEE National Aerospace and Electronics Conference (NAECON)

Abstract

We summarize the history and state of the art in CNNs, which constitute a significant advancement in pattern recognition. As a demonstration of capability, we address the problem of automatic aircraft identification during refueling approach. In this paper we describe the history of CNN development and provide a high level overview of the state of the art and a summary of leading CNN libraries with Compute Unified Device Architecture (CUDA) support. Finally, we demonstrate an application of CNN technology to autonomous aerial refueling and identify areas of follow-on research.

Introduction

Recent emergence of high performance deep learning toolkits and libraries such as Caffe[11], TensorFlow[15], and Theano[16] have spurred advances in deep learning research as summarized in [4]. In particular, the ImageNet competition of the last 5 years has produced a number of meaningful advances to convolutional neural network architectures and techniques that make this a feasible technology for a variety of object recognition applications.

To give the reader a sense for the considerable impact of this burgeoning field, this paper traces its roots to biological origins and early development prior to the

ImageNet competition, describing major issues that hindered earlier success and how they have been overcome. We then describe the general structure of a CNN, illustrated with a tour of the LeNet-5 framework, as a pedagogical example of modern implementations. The most recent advances that constitute the current state of the art are manifested in AlexNet which contextualizes developments in deep learning theory and practice. The tutorial section of this paper concludes with a description and comparison of the leading CNN libraries.

As a case study that shows the potential for CNNs in a specific setting, we show an illustrative example of aircraft identification for aerial refueling. We describe the issues in training and classification using CNNs from a 76,000 image dataset. We implement a deep learning classifier in Python by employing Berkeley Caffe[11] resulting in an AlexNet[12] style classifier with 10 classes of military aircraft. The case study allows us to describe important issues with training data and classifier integration that sets the foundation of future study.

CNN Development Chronology

The promising performance of early CNNs, inspired by the mammalian visual system, has greatly improved in recent years due to developments in general Artificial Neural Network (ANN) training algorithms, improvements in CNN structure, inexpensive access to high performance computation, and availability of large training datasets.

Mammalian Visual System.

In the late 1960s, Hubel et al.[17] investigated the functional architecture of the visual processing pathway in felines and primates identifying a hierarchical laminar structure. Hubel's methods included electrical probing of neuronal activation in the

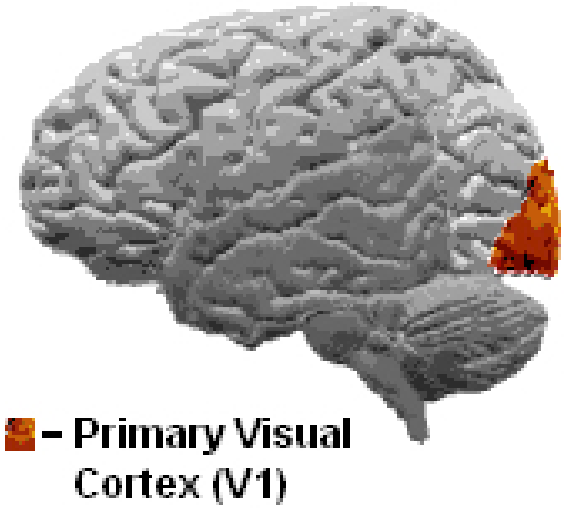


Figure 1. The Striate Cortex, a.k.a. the V1 portion of the visual pathway (shown in orange.)

striate cortex (see Fig. 1) while exposing the eyes of lightly anesthetized subjects to various patterns of visual excitation. Contrasting sharp lines and repeated bar shapes with controlled angular orientation were typical, as well as uniform and varied color fields.

Early CNNs.

The field of CNNs is pre-dated by early experiments [18] with hard-wired, analog neural networks ranging back to the 1940's. By the 1970's researchers [19] attempted to simulate ANNs on digital computers but were hampered for several reasons. The meager computational resources of the time simply could not support experimentation with ANNs with more than a few neurons. Also, no rigorous training methodology existed for ANNs with more than one or two layers. Lastly, a paucity of labeled data prior to the advent of the world wide web [20] greatly increased the cost of ANN research. These issues, in turn, hampered CNN development.

Fukashima [21] was inspired by Hubel's work on the Mammalian visual pathway

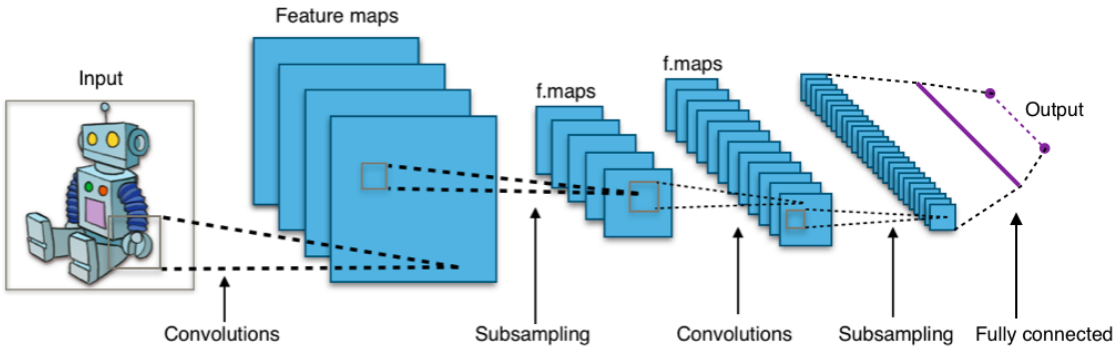


Figure 2. Typical Architecture of a LeNet-5 like Convolutional Neural Network. Each plane is a feature map. i.e. a set of units whose weights are constrained to be identical.

when experimenting with the Cognitron and then the Neocognitron. His work accomplished shift invariant pattern recognition via ANN for the first time, but suffered from a relatively ineffective ad-hoc training methodology and modest computational means of the late 1970's.

ANNs are trained by making use of Stochastic Gradient Descent (SGD) to seek out approximately optimal neuron weights appropriate for pattern recognition. SGD requires gradients of pattern recognition error with respect to each network parameter. The back propagation chain derivative technique which provides these gradients, was first applied in the ANN research community in 1984 by Werbos [19][22][23]

Fukashima's work was followed by others in the 1990's and by 1998, LeCun developed a check-reading CNN, LeNet-5[24][25] similar to Fig. 2. This network is viewed as a major advance in CNN in performance, taking advantage of faster computational hardware and the back propagation gradient computation.

In 2007 Serre [26] demonstrated robust, biologically inspired image recognition performance matching or exceeding that of traditional image processing techniques such as Scale Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG). Serre trained CNNs to recognize dozens or hundreds of image classes (Caltech 5, Caltech 101, MIT-CBCL) after training on thousands example images.

Modern accomplishments.

Since Serre's work in 2007, standardized visual object recognition challenges have driven the state of the art in image classification. The PASCAL Visual Object Classes (VOC) challenge has been running since 2005 and was superseded by the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [12] [5] since 2010. Imagenet ILSVRC, a collaboration of web-commerce companies and academia, curates large databases of labeled imagery free for use by researchers at large. As of 2016, Imagenet reposes 15 million labeled images in approximately 22,000 categories[5].

By utilizing General Purpose Graphics Processing Units (GP-GPUs) and the 2010 ILSVRC dataset, Hinton and Krizhevsky [12] built and tested what is considered the first modern large scale CNN and set the de facto standard to which new CNNs are compared. In the ILSVRC 1000 class image recognition competition, top 1 and top 5 are test set classification accuracy metrics. Top 1 is the percentage of the test set properly classified by the CNN's highest activation output class. Top 5 takes into account the five highest activation classes. Prior to Alexnet, the best performance was achieved by sparse coding methods with 47.1% and 28.2% classification error, and SIFT methods achieved 45.7% and 25.7% classification error. Alex-net achieved top 1 and top 5 error rates of 37.5% and 17.0% respectively in the 2010 ILSVRC competition, marking the emergence of CNNs as the dominant technique in image classification.

Subsequently, Google's 22 layer 'GoogLeNet' classifier [27] achieved 6.67% top 5 classification error on the ILSVRC 2014 data set. For context, the ILSVRC organization estimates the top 5 classification error of trained human subjects to be approximately 5.1% [5]. Test subjects spent significant time to become familiarized with the 1000 ILSVRC class labels, then spend approximately one minute classifying each image.

In 2015, He, et al [28] achieved 4.94% top five classification error in the ILSVRC 2014 dataset by modifying the Rectified Linear Unit (ReLU) basis function. This work is the first claim of CNN performance matching or exceeding that of humans on an ILSVRC dataset. Similarly, Sermanet utilized a CNN to achieve 94.85% accuracy in reading real-world house numbers in images taken from Google Street View. In this task, humans typically perform at 98% accuracy[29]. The trend in CNN development is toward increasingly deep networks, with performance matching that of the average human for specific pattern recognition tasks.

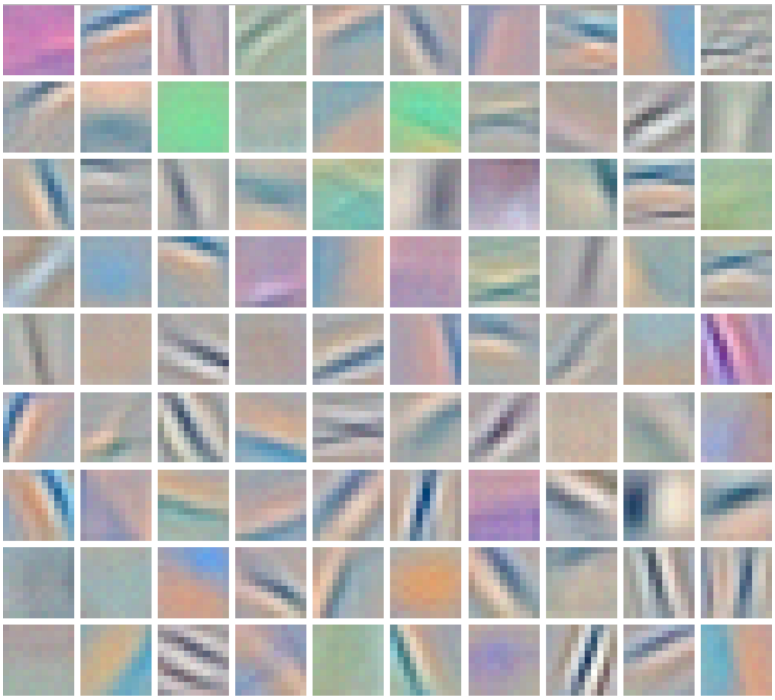


Figure 3. First layer convolutional features of AfCaffe, an Alexnet derivative.

CNN description

All CNNs possess a common generic structure. As detailed in Fig. 2 and Fig. 4, CNNs are typically composed of a multi-layer convolutional portion, which takes imagery as input and outputs high level features to a fully connected Multi Layer

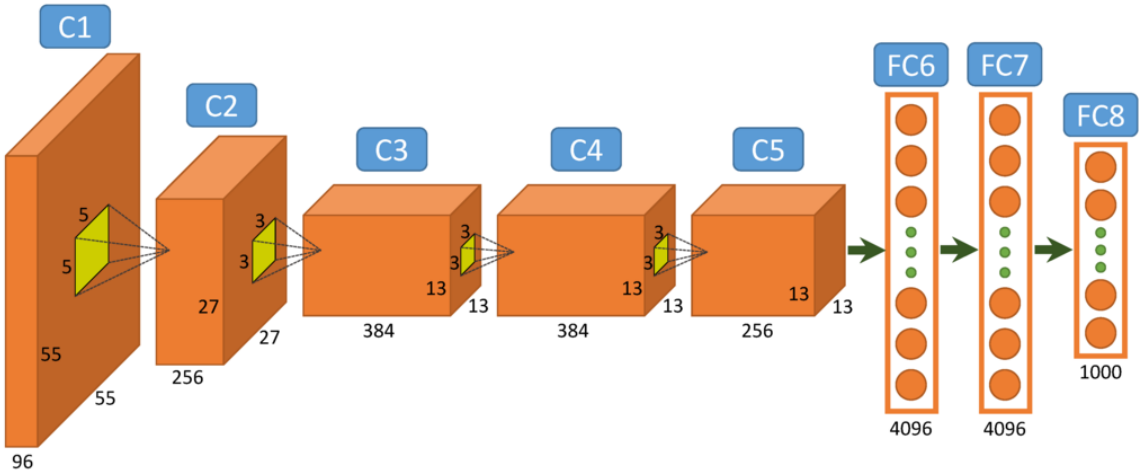


Figure 4. Architecture of Alexnet. From left to right (input to output) five convolutional layers with Max Pooling after layers 1,2, and 5, followed by a three layer fully connected classifier (layers 6-8). The number of neurons in the output layer is equal to the designed number of output classes[30].

Perceptron (MLP) classifier. The entire network is trained end-to-end via SGD which minimizes classifier output error.

Convolutional Layer.

The convolutional portion is typically composed of alternating convolutional and pooling layers. As depicted in Fig. 2, a convolutional layer is a collection of feature maps, each composed of identical neurons with several inputs and a single output. However, unlike the fully connected MLP, each neuron in a CNN layer is connected to a relatively small number of inputs [21][20].

These inputs can be viewed as rectangular kernels, or local receptive fields [24], regularly arranged over the input image with partial overlap. Critically, the weights of these neurons, one for each pixel of the kernel, are shared over each neuron in the feature map. As a consequence of weight sharing, each of the convolutional kernels in a feature map are identical. Therefore, the outputs of the neurons which compose a feature map all correspond to the response of a single repeated kernel 'slid' around the input image. This operation is equivalent to mathematical convolution: (slide,

multiply, sum).

A feature map then, outputs a matrix of scalars that map the presence of a feature over the input matrix. Because the kernels overlap, if a kernel feature is present in the input image, and is scaled to fit inside of a kernel, then it will be represented in one or more adjacent locations in that feature map. In this way, overlapping convolutional kernels contribute some amount of distortion, shift, scale, and rotation invariance.

Pooling (Sub-Sampling).

Depending on the number of feature maps in a convolutional layer, the output in terms of memory footprint can be much larger than the input size. A pooling layer is often used after each convolutional layer to reduce resolution and as a by product, increase invariance to translation.

Typically, a feature map's resolution is reduced by a factor of 2 or 3 in each dimension. That is, the input feature map is divided up into kernels of size 2x2 or 3x3, and a pooling operation is performed on the scalar values presented by the feature map to each kernel. Mean value pooling was typically used prior to 2007[26], but use of the max() operator has been shown empirically to be much more robust [12][29][31]. In vector analysis terms, if the kernel inputs are arranged as a vector $\overline{a} = [a_1, a_2, \cdots, a_n]^T$ then the maximum operator is equivalent to the max norm

$$||\overline{a}||_\infty = \max\left(|a_1|, |a_2|, \cdots, |a_n|\right), \tag{1}$$

where the output is simply the magnitude of the largest entry in $\overline{a}$. In this way, max pooling preserves the magnitude of the largest feature response presented to it by the feature map of the previous convolutional layer. An additional consequence of pooling is an efferent reduction in spatial resolution with increase in the number of features per layer.

Fully Connected Classifier.

The output of the convolutional portion of the CNN is essentially the low resolution spatial positions of a large number of abstract features. The classifier portion is typically a fully connected MLP, which recognizes patterns of features that correlate with known output classes. The output layer of the classifier portion has precisely the number of outputs as there are class labels in the input data set. An optional Softmax layer can be added to normalize the outputs to a probability mass function, depending on design requirements.

CNN backpropagation.

The back propagation technique is commonly used to evaluate gradients of global classification error with respect to neural network parameters (i.e., neuron weights and bias values) in order to utilize SGD. Back propagation requires only minor modification [24] to accommodate the shared weights configuration of CNNs.

State of the Art in CNNs

To be useful, modern CNNs have large capacity and depth, and are consequently prone to overfitting. To avoid this, CNNs must be trained with a large amount of labeled data. ReLU activation functions, dropout parameter regularization, and data augmentation are techniques used to cope with large datasets, and improve CNN performance.

Depth.

A fundamental property of neural networks is the relationship between depth and abstraction. Generally speaking, the knowledge represented by features in each layer of a neural network increases in abstraction with depth. For example, the layer one

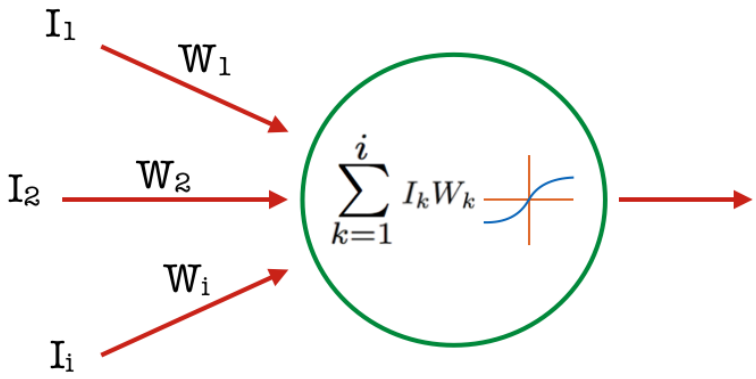


Figure 5. Artificial neuron diagram. Inputs on the left are multiplied by their associated weights (learned values). The sum of these weighted inputs is applied to an activation function which computes the neuron's output.

features represented in Fig. 3 are essentially Gabor filters, representing simple patterns like edges and orientations. However, in higher levels in the network, features are increasingly abstract, culminating in the final layer where classes of objects are represented. Krizhevsky [12] and LeCun, et al. [4] have shown that depth is essential in practical, high capacity neural networks. Indeed Bengio [32] has shown that invariance to distortion and increased selectivity comes with increasing depth.

ReLU Activation Function.

The essential utility of neural networks is their ability to approximate, or learn, many of the highly non-linear functions presented by the natural world. Central to this ability is the activation function for the neurons in the network as diagrammed in Fig. 5. Several activations functions are pictured in Fig. 6. Of particular import are the sigmoid and ReLU.

A neuron's activation function must be non-linear in order to capture non-linear relationships between its inputs. Also, in order to be trained via gradient descent, activation functions must be differentiable. The sigmoid activation function, as shown in Fig. 6, is continuously differentiable but flat in the extremes. This leads to a

problem of vanishing gradients when the function is strongly activated. The effect of small gradients is prohibitively long training times and in deep networks, numerical accuracy issues.

The ReLU [33] activation function was developed in response to this problem with great success. The ReLU computes the Maximum(0,x) function where x is the sum of weighted inputs. Note that while ReLU is not strictly differentiable when x equals precisely zero, this is not a practical problem with floating point arithmetic, as it is differentiable when x is arbitrarily close to zero. There are some open questions posed in [34] as to why ReLU speeds up neural network training as much as it does, but nonetheless it is used almost exclusively[4] in modern CNNs.

Dropout Regularization.

Another indispensable feature of state of the art CNNs is training data dropout which greatly reduces the propensity of ANNs to over fit a training data set. During training, one half of the neurons in the fully connected classifier portion are randomly omitted from SGD update. The effect of which is to prevent co-adaptation of adjacent layer feature detectors to specific combinations of features. Instead, the layers are trained to find generally useful feature combinations rather than specifically those found in the training set [35]. Dropout regularization provides large gains in classification accuracy on held-out test imagery [2].

Training Data Augmentation.

Data augmentation randomly applies certain types of transforms to training images and is a critical over fit mitigation technique used in modern CNNs. Typical transformations include random cropping, mirroring, rotations, affine, and color transformations. The effect of which is to increase the size of the training data set by

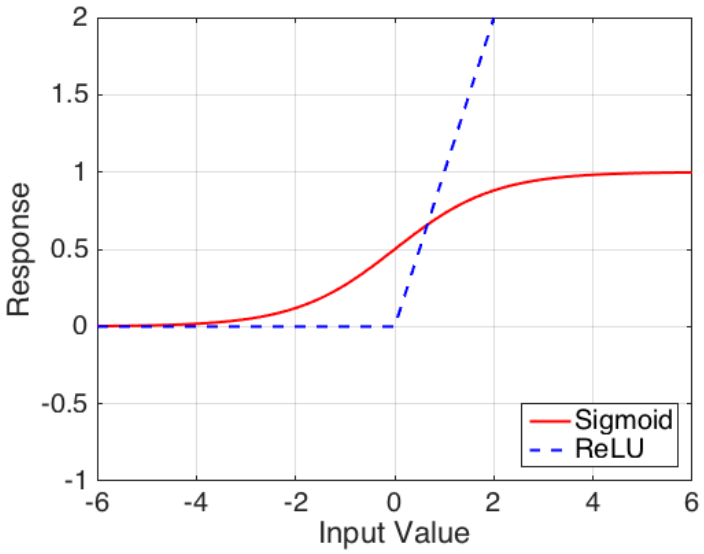


Figure 6. Sigmoid and ReLU activation functions.

a large factor, beyond the capacity of the CNN to memorize. This forces the CNN to recognize general features in the training set. Random cropping and mirroring enjoy wider usage than more computationally demanding transformations such as rotations and affine transformations [36] [37] [38].

Limitations.

At present, state of the art CNNs have several limitations. In terms of performance in image recognition, CNNs can be trained to a level of performance on par with humans [33]. However, the human visual pathway is able to generalize over real-world imagery with far fewer observations than CNNs. That is, the training set size required for CNNs is much larger than that required for humans. How the brain accomplishes this is an open question and an area of active research in the field of Generative Adversarial Networks [39].

The need for large datasets presents a practical problem for development of CNNs. The Imagenet [5] organization and others attempt to address this problem with some success. However, in domain specific applications, manual data collection efforts are

often required.

After the training phase is complete, an interestingly large CNN typically requires several dozen milliseconds to perform a forward pass on a commodity CPU and requires tens of megabytes of storage for its network parameters. These requirements effectively limit deployment to compute or power limited applications such as mobile devices.

Implementations

Various libraries exist for implementing neural networks. These libraries are designed with different philosophies and therefore have different approaches. For example, Caffe seeks to separate implementation from network design and encourages the use of configurable predefined layers, whereas Torch seeks to be a high-performance library for implementing and extending numerical algorithms.

Caffe is an open-source library developed in C++ by Berkeley Vision and Learning Center. It supports Python and Matlab interfaces. The goal is to separate the network design from its actual implementation to allow easier creation of networks. It supports CUDA and is one of the most popular libraries and has been widely used.

Neon is a relatively new open-source library written in Python by Nervana. The CPU and GPU backends are custom built libraries also developed by Nervana. Nervana claims that their GPU backend has 2x the performance of cuDNNv4.

TensorFlow is an open-source library developed in C++ by Google. The library has support for Python and C++ interfaces. Google uses it internally to power some of their search technologies. It uses data flow graphs for performing numerical analysis by representing nodes as operators and edges as numerical data arrays that connect the nodes. As of the authors writing, it supports cuDNNv2 (instead of cuDNNv3), which is attributed for its poor performance in GPU tests.

Theano is an open-source library written in Python by the University of Montreal. It is both written and interfaced with Python. It allows for symbolic expressions and performs auto differentiation, an important feature for implementing gradient descent. It uses CUDA as its backend for GPU-acceleration. Although it is technically not a dedicated deep learning library, it is widely used for this purpose. The authors consider it one of the most extensible libraries available.

Torch is another open-source library written in C with a LUA interface. It was created and is maintained by Ronan Collobert and Clement Farabet. It also has the ability to perform automatic differentiation based on the network architecture (rather than symbolic equations). It is popular with and receives support from large organizations such as Facebook, Twitter, and NYU.

In [40] a number of different deep learning libraries are compared and benchmarked. Included are Caffe, Neon, TensorFlow, Theano, and Torch. Both training times and forward pass times are evaluated. Libraries are tested using a LeNet implementation on the MNIST data set and the AlexNet implemenation on ImageNet dataset on CPU and GPU. The results show that Torch achieves the best results in almost all cases. Theano performs well on MNIST but poorly on AlexNet. TensorFlow consistently has the worst GPU performance due to its use of an older CUDA library. In terms of use, [40] finds Torch and Theano are the most extensible, although they note that Torch has inferior documentation.

Case Study: Vision System for Tanker Aircraft

Visual identification of an aircraft by model is an object classification problem well suited to CNNs. Consider a future scenario involving autonomous refueling tanker aircraft operations. Automated systems may need to visually identify aircraft on approach in order to configure avionics systems for formation flight and to set up fuel

transfer systems.

A typical engineering approach to applying recently developed CNN technology is to adapt a successful network design to a specific engineering problem. Utilizing this technique, Alexnet was adapted and trained to classify aircraft imagery into ten pre-defined training categories: A-10, B-1, B-2, C-17, C-130, CV-22, F-15, F-16, F-22, and F-35.

Figure 7. Refueling Approach Diagram[41].

Figure 8. Refueling approach camera perspective.

Fine Grained Classification.

Alexnet was originally designed to compete in the 1000 class ILSVRC, therefore, the output layer, as seen in Fig. 4, has 1000 output neurons, each representing an

output class. The number of neurons in the output layer was reduced to ten, to match the design of a ten output classifier. Furthermore, as the network only needs to recognize ten discrete classes, the capacity of the Alexnet classifier design is larger than is required for our problem, providing a degree of design freedom in optimizing our CNN to the design problem. Eight variations of classifier hyper-parameters, i.e., the number of fully connected neurons in layers six and seven, were evaluated as shown in Fig. 12 and Table 1. The convolutional portion of the network (layers 1-5) was retained from the Alexnet configuration. The resulting optimal CNN is referred to as AfCaffe.

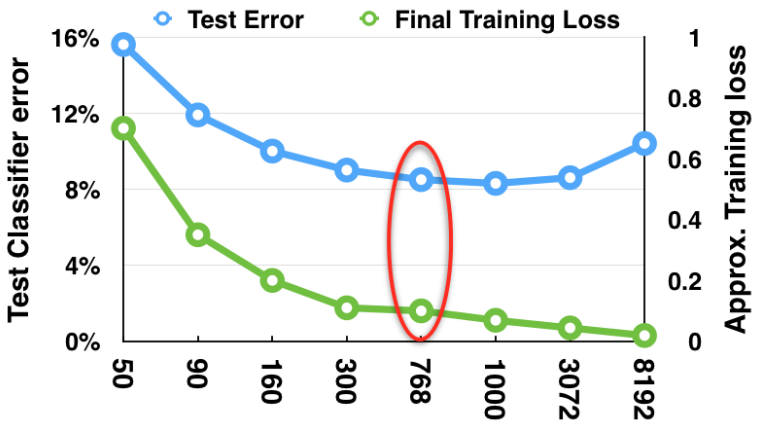


Figure 9. Test and Training error vs Classifier hidden neurons.

Table 1. Test and Training error vs Classifier hidden neurons.

Total Classifier Hidden Neurons	Test Accuracy	Test Error	Final Training Loss	Layer 6 Neurons	Layer 7 Neurons	Layer 8 Neurons	snapshot size (MB)
50	0.844	15.6%	0.7	30	20	10	10.3
90	0.881	11.9%	0.350	60	30	10	11.6
160	0.90	10%	0.2	100	60	10	13
300	0.91	9%	0.11	200	100	10	16.8
768	0.915	8.5%	0.1	512	256	10	29
1000	0.917	8.3%	0.069	700	300	10	36
3072	0.914	8.6%	0.044	2048	1024	10	93
8192	0.896	10.4%	0.019	4096	4096	10	227

A body of imagery consisting of 76,426 images was sourced from the internet,

primarily from Flickr groups and refueling sequences captured from You-Tube videos. These data were selected to roughly evenly represent each aircraft class. Each image was given a numeric 1-10 label representing the class of aircraft captured in the image. Next, the dataset was randomly divided into two subsets, training and test. The training subset consisted of 61342 images (80%) and the test set 15084, (20%).

Validation.

Inspection of the learning curves detailed in Fig. 12 indicates a bias vs variance trade-off relationship between test error (the compliment of classification accuracy) and the capacity of the classifier portion of the network. The optimal point with minimum test error occurs with 1,000 total hidden neurons in the classifier portion of the network.

Conclusions and Future Work

Individual deep CNNs or ensembles thereof represent the state of the art in image recognition since 2012, demonstrating human level performance in specific domains. This thesis summarized the biological inspirations and historical development of CNNs followed by detailed descriptions of basic and modern CNN operation in *CNN Description* and *State of the Art in CNNs* and provided an overview of current deep learning toolkits.

AfCaffe, a CNN developed in the *Case Study* section, was successfully trained to perform fine grain aircraft recognition of ten models of United States Air Force (USAF) aircraft, largely independent of perspective. Optimization of classifier test performance was accomplished by analyzing the bias-variance trade-off relationship between test set performance and classifier size.

Future work on the aerial refueling application example may involve driving classi-

fication accuracy toward human norms. This work is expected to take several fronts: (1) Utilizing training data augmentation techniques outside the scope provided by the basic Berkeley Caffe library. (2) Exploring ensemble techniques [4]. (3) Develop a human level performance metric by quantifying the norms of aircrew performance in classifying aircraft from video imagery.

III Scholarly Article: Improved Aircraft Recognition for Aerial Refueling through Data Augmentation in Convolutional Neural Networks

Proceedings of the 2012th International Symposium on Visual Computing

(ISVC 2016)

Robert Mash, Brett Borghetti PhD, John Pecarina PhD

Abstract

As machine learning techniques increase in complexity, their hunger for more training data is ever-growing. Deep learning for image recognition is no exception. In some domains, training images are expensive or difficult to collect. When training image availability is limited, researchers naturally turn to synthetic methods of generating new imagery for training. We evaluate several methods of training data augmentation in the context of improving performance of a Convolutional Neural Network (CNN) in the domain of fine-grain aircraft classification. We conclude that randomly scaling training imagery significantly improves performance. Also, we find that drawing random occlusions on top of training images confers a similar improvement in our problem domain. Further, we find that these two effects seem to be approximately additive, with our results demonstrating a 45.7% reduction in test error over basic horizontal flipping and cropping.

Introduction

Training Data augmentation and simulation have become standard tools used for training practical deep neural networks in situations where there is not a very large amount of training data available. The popularity of Krizhevsky's Alexnet as a model CNN has led to adoption of his computationally simple Data Augmentation (DA)

techniques as a default. Krizhevsky's strategy applies random horizontal flipping and randomly located fixed-size-cropping to each training image. As these amount to matrix indexing operations, they are computationally simple, but don't take full advantage of DA's potential. The negative consequence of using only simple DA techniques, is that practitioners tend to expand their network size for a given test set performance. The techniques we explore improve test set performance without increasing down stream network size and subsequent computational complexity.

In Related Work we discuss machine learning norms in data augmentation and simulation. In Problem Domain, automatic visual aircraft recognition is introduced in the Automatic Aerial Refueling (AAR) domain.

In Methodology, the AfCaffe CNN from [42] is introduced in the AAR context and we detail several types of data augmentation. We conclude that significant performance improvements on held out test data can be made by applying these DA techniques individually or in combination. Using combinations of DA techniques, we demonstrate a 45.7% reduction in test set error on our aircraft recognition dataset. Also, we expect that due to the high cost of gathering aerial photography, AfCaffe may be further improved using simulated training data in the future.

Related Work

Training Data Augmentation.

A critical tool for overcoming training data overfit with modern CNNs is data augmentation, the use of randomly applied data transformations which greatly expand the effective size of a training set. Data augmentation randomly applies certain types of label preserving transforms to training data. Typical transformations include random cropping, mirroring, rotations, occlusions, as well as affine and color transformations. The effect of which is to increase the size of a training data set by a large

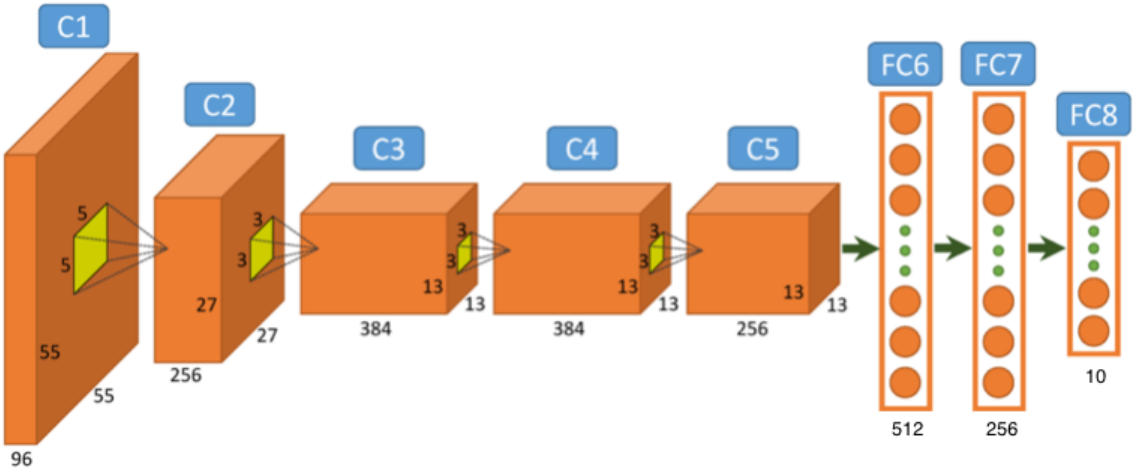


Figure 10. Architecture of AfCaffe[42], an Alexnet[12] derivative. From left to right (input to output) five convolutional layers with Max Pooling after layers 1, 2, and 5, followed by a three layer fully connected classifier (layers 6-8). The number of neurons in the output layer is equal to the designed number of output classes.

factor, beyond the capacity of an ANN to memorize, or over-fit. The network is then forced to learn hierarchical feature detectors useful for general pattern recognition.

Data augmentation normally includes randomly applied mirroring and random cropping as in [12][43][44][45]. When other methods of data augmentation are used, they are typically applied in addition to this baseline as in [14]. Here Dielman applies random rotations, scaling, and brightness jittering in addition to randomly applied mirroring to images of galaxies.

This is an example of a combination of data augmentation methods only suitable to a certain domain. In the case of galactic images, which are often rotation invariant, unconstrained random rotations can be applied resulting in images still recognizable as galaxies. In the case of aerial refueling images as in Fig. 11, the refueling boom occludes the image of the approaching to-be-refueled aircraft. Consequently, randomly drawn occlusions are an appropriate data augmentation method in this domain and others as explored in [46]. Limited angle random rotations of non-rotation invariant objects can sometimes be effective as in [47][37]. However, others have cited null results as in [48].

Synthetic Training Data.

A possible source of neural network training data are simulated images from the field of computer graphics as in [49]. Much work has been done in the sub-fields of vehicle and pedestrian detection as in [50], [51] and [52]. Synthetic datasets enjoy several benefits over manually acquired photographic datasets. Instances of training data can be generated with total control over perspective, texture, lighting, range, orientation, etc. as in [53]. Further, in some fields of inquiry, real-world photography can be much more expensive to acquire than synthetic data. Lastly, training data labels can be generated with perfect accuracy, a particularly useful attribute when working in a continuous domain such as position and orientation regression.

Generative Networks.

A new and burgeoning field is that of generative adversarial networks (GANs)[54] used to 'imagine' variations on a training image, based on the types of variations learned from real-world imagery. Two main types of have been developed, the Deep Convolutional GAN or DC-GAN[55] and the Laplacian - GAN or Lap-GAN[39]. For example, in a recent cross-over between these related fields Dosovitskiy in [56] trains a convolutional GAN to generate photorealistic images of many different never-seen-before chairs from previously rendered images of chairs from simulation.

Problem Domain

Consider a future scenario involving autonomous aerial refueling tanker aircraft operations. Visual identification of an aircraft by model is currently performed by trained aircrew. However, recent success in aircraft image recognition with AfCaffe in [42] motivates an approach wherein a CNN automatically identifies the model of an approaching aircraft *visually*, in lieu of a human crew member.

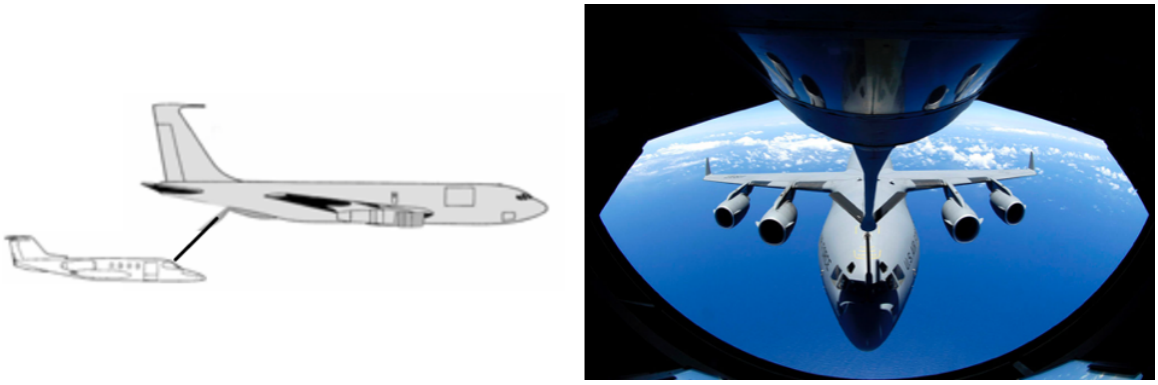


Figure 11. (left) Refueling Approach Diagram[41] (right) Refueling approach camera perspective.

In an autonomous refueling tanker, visual recognition of approaching aircraft is but one part of a greater collection of autonomous systems. For example, after recognizing the model of aircraft, the position and orientation of the aircraft must be continuously estimated in support of the tight formation flying required in the domain.

AAR Vision System Concept of Operations.

During aerial refueling, an aircraft to be refueled approaches a tanker aircraft from below and behind in a predefined flight path resulting in the relative positions shown in Fig. 11. A refueling tanker equipped with an AAR vision system would have rear-facing cameras aligned with the aircraft approach vector. This would result in a forward perspective of an aircraft on refueling approach as shown in Fig. 11.

The refueling perspective imagery would then be down sampled as appropriate to match the input spatial and color dimensions of a previously trained CNN. Next, a forward pass of the CNN would be performed, the output of which is a probability mass function estimating the model of the approaching aircraft.

The aircraft model estimate could then be used for configuration of fuel transfer systems, formation flight systems, and generally improved situational awareness.

Methodology

Aircraft Recognition with the AfCaffe CNN.

AfCaffe, as shown in Fig. 10, is a derivative of Alexnet which was originally designed to compete in the 1000 class ILSVRC in [12]. The number of neurons in the output layer was reduced to ten, corresponding to ten models of U.S. Air Force aircraft. Further, as AfCaffe only needs to recognize ten classes, the capacity of the Alexnet classifier design was optimized by reducing the number of neurons in the fully connected classifier. As seen in Fig. 12, eight variations of classifier hyper-parameters were evaluated with size 768 providing the best performance to capacity ratio. The convolutional portion of the network (layers 1-5) was retained from the Alexnet configuration. As AfCaffe has 7.8 million trainable parameters, it requires a large number of training examples in order to prevent overfitting to the training dataset.

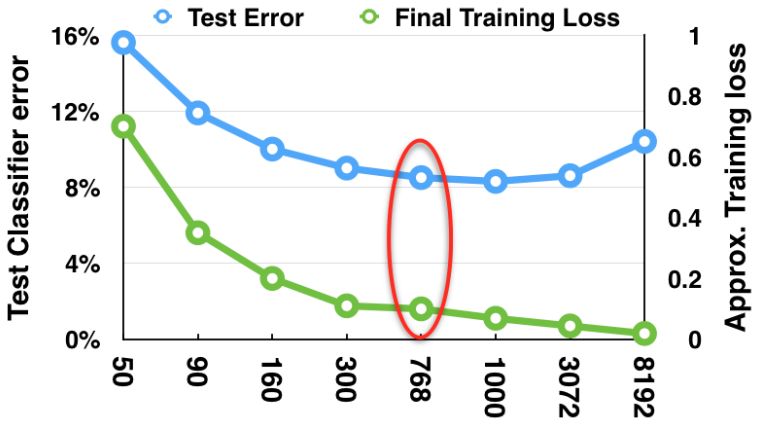


Figure 12. Test and Training error vs Classifier hidden neurons from [42].

Training and Test Data.

In order to train and evaluate AfCaffe, a body of imagery consisting of 76,426 images was sourced from the internet, primarily from Flickr groups and refueling sequences captured from YouTube videos. These images were selected to roughly evenly represent each aircraft class. Each image was given a numeric 1-10 label representing the class of aircraft captured in the image. Next, the dataset was randomly divided into two subsets, training and test. The training subset consisted of 61,342 images (80%) and the test set 15,084, (20%).

The training set input images are stored in a Lightning memory Mapped DataBase (LMDB) at 256x256 pixels with three 8-bit color channels. During neural network training, images are extracted from the LMDB and randomly transformed while preserving associated class labels. After transformation according to one or more of the following techniques, the images are applied to the input layer of the AfCaffe CNN which accepts images of resolution 227 x 227 with three 8-bit color channels.

Data Augmentation Techniques .

Several types of DA are evaluated as shown in Fig. 13. They are described in detail as follows.

- Baseline: No data augmentation is performed on the input image. The input image is simply down sampled in order to match the 227x227 CNN input resolution.

- H+RC: The input image is flipped horizontally with probability = 1/2. This operation is followed by a fixed size crop randomly applied to the input image. The fixed crop size corresponds to the input dimensions of the CNN. Krizhevsky, et al. applied this technique with great success in [12]. Both the

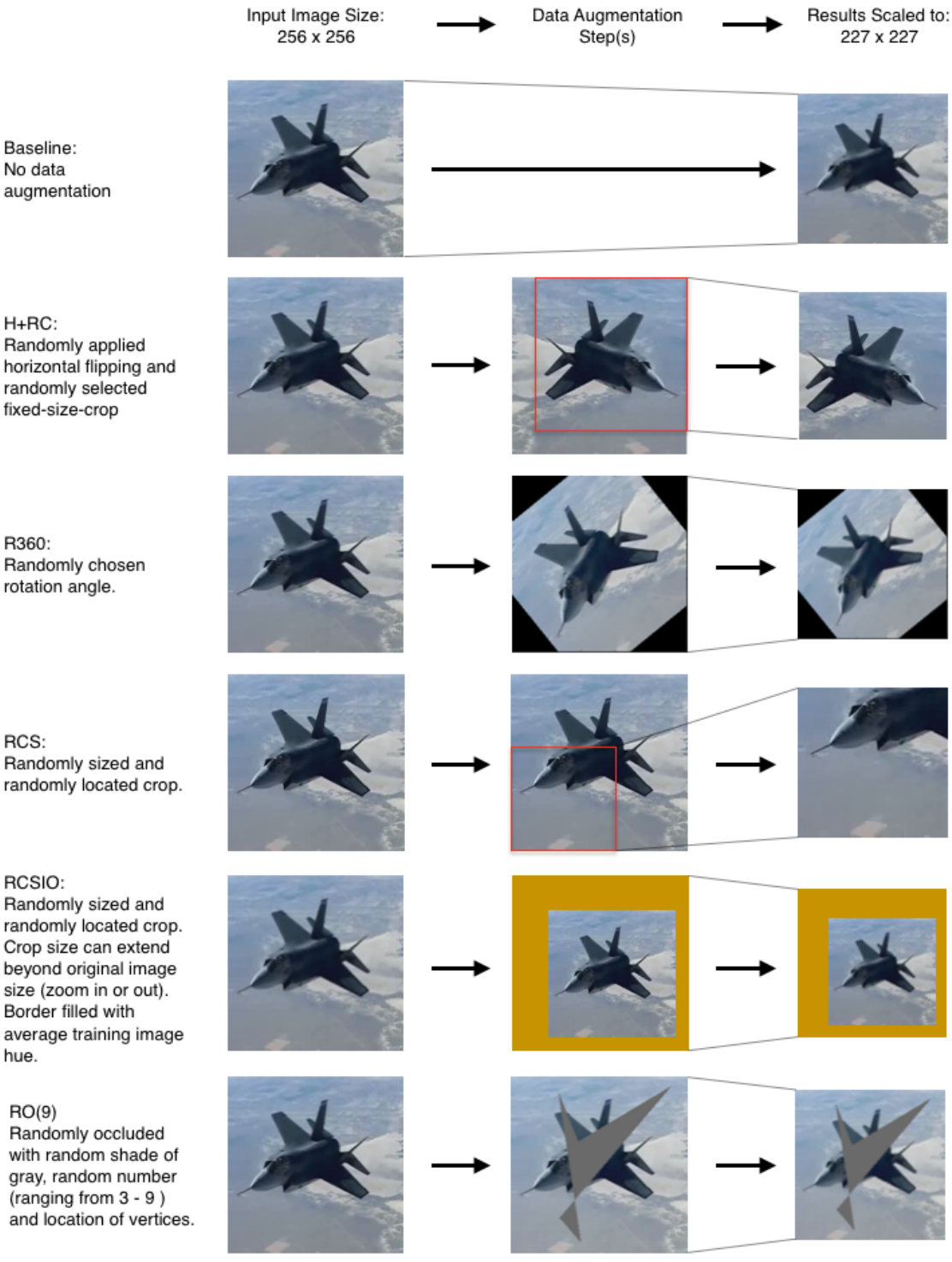


Figure 13. Data Augmentation Types.

flipping and cropping operations amount to matrix indexing, which makes H + RC computationally inexpensive to perform during training.

- R360: The input image is rotated about its center by an angle randomly chosen from a uniform distribution ranging from 0 to 360 degrees. As seen in Fig. 13, black triangular regions are left in the corners of the image after the rotation is performed. The resulting image is then re-sampled as appropriate to match the CNN input resolution.

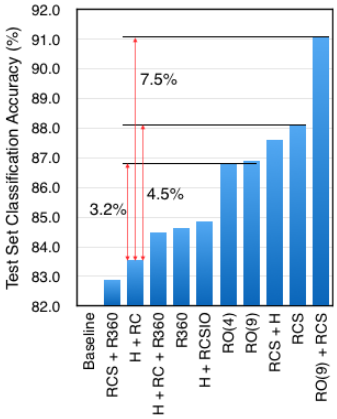
- RCS: A randomly sized and randomly located crop is taken from the input image. The square crop dimension is randomly chosen from a uniform distribution ranging from 150 pixels to 256 pixels. The crop location is then randomly selected from uniform distributions across the input image's x and y dimensions. The resulting crop is then re-sampled as appropriate to match the CNN input resolution. An example is shown in Fig. 13.

- RCSIO: The input image is scaled by a factor ranging from 0.39 to 1.13 randomly selected from a uniform distribution. Then a fixed size crop of 227 x 227 is taken from a location randomly selected from a uniform distribution on the resulting image. If the selected scale factor is less 1.0, the input image is effectively scaled down in size, resulting in a margin or border around the edges of the image as seen in Fig. 13. The resulting border is then filled with the mean training image hue to keep from shifting the mean of the training set. The scaled image is then re-sampled as appropriate to match the CNN input resolution.

- RO(n): Random occlusions are applied to the input image by first selecting a random number of vertices ranging from 3 to n, where n $>$ 3. Locations for the selected number of vertices are then randomly selected from uniform distributions over the input image. The vertices are then used to draw an

irregular polygon over the image as seen in Fig. 13 where the fill color is chosen as a randomly chosen shade of gray. The resulting image is then re-sampled as appropriate to match the CNN input resolution.

- Combinations: Various combinations of the approaches above are possible. H + RC + R360, RCS + R360, and RO(9) + R360 are implemented in this study. In each case, the transformations are performed in left-to-right reading order followed by resampling to match the CNN input resolution.



Data Augmentation Type	Description Augmentation = Randomly Applied:	Final Training Set Loss	Test Set Classification Accy (%)	Test Set Classification Error(%)
Baseline	No Data Augmentation	0.4	82.0	18.0
RCS + R360	Random Crop Size + Rotations	0.41	82.9	17.1
H + RC	Mirroring + Fixed Size Crop	0.4	83.6	16.4
H + RC + R360	Mirroring + Fixed Size Crop + Rotations	0.4	84.5	15.5
R360	Rotations	0.4	84.6	15.4
H + RCSIO	Mirroring + Random Crop Size with Zoom In and Out	0.39	84.8	15.2
RO(4)	4 Vertex Polygonal Occlusions	0.36	86.8	13.2
RO(9)	9 Vertex Polygonal Occlusions	0.36	86.9	13.1
RCS + H	Random Crop Size + Mirroring	0.41	87.6	12.4
RCS	Random Crop Size	0.36	88.1	11.9
RO(9) + RCS	9 Vertex Polygonal Occlusions + Random Crop Size	0.38	91.1	8.9

Figure 14. Test set classification accuracy vs Data Augmentation techniques. Early stopping at training loss = approximately 0.4 for all experiments.

Implementation.

The AfCaffe CNN was developed using the Caffe Deep Learning Framework from the Berkeley Vision and Learning Center. The following Caffe specific training parameters were used: base learning rate=0.001, gamma=0.1, stepsize=7500, momentum=0.9, weight decay = 0.0005, batchsize = 250

Data augmentation techniques were implemented using the Sk-Image and OpenCV libraries in Python. Due to time constraints, no cross validation has been performed.

Results

As seen in Fig. 14, Krizhevsky style mirroring and fixed size random cropping (H + RC) provides some benefit over the baseline case on which no data augmentation is performed. This is not suprising, as the baseline case is subject to significant over-fitting of the training set.

Transitioning from a fixed crop size randomly selected from the original image to a randomly selected crop size (RCS) improves overall test set classification accuracy by 6 percentage points over the baseline and 4-5 points over Krizhevsky's H + RC method. Conceptually, this is likely due to increasing scale invariance, as RCS applies the same training images to the CNN at various scales.

Additionally, in the aircraft recognition domain, random polygonal occlusions yield a similar improvement over the baseline and Krizhevsky's H + RC method as from random scaling. It's not clear if this is a domain specific improvement where some test aircraft images are occluded by a refueling boom as in Fig. 11. For example, Yilmaz concludes in [46] that random occlusion improves classification performance with Recurrent Neural Networks (RNNs), lending some credibility to the idea that randomly drawing occlusions on training images may improve the generalizing capability of ANNs. It may be that random occlusions contribute a drop-out like parameter regularization effect concentrated in regions of the image most represented by the probability distribution controlling the random occlusion generator.

Lastly, as detailed in Fig. 14, the greatest improvement in test set classification accuracy occurs when these two techniques are combined. Random occlusions (RO) and randomly selected crop size (RCS) seem to contribute almost independently to generalization as the total improvement over Krizhevsky's method is approximately additive.

Conclusions

Applying more computationally expensive DA techniques such as random crop scaling (RCS) and random occlusions (RO) to a training set improves CNN performance significantly over simpler DA methods as pioneered by Krizhevsky. By combining RCS and RO, we demonstrate a 45.7% reduction in test set classification error over Krizhevsky's method.

The primary benefit of more computationally expensive DA techniques is an increase in test set classification performance without an increase in network size and downstream computational complexity. The only cost of these DA techniques is an increase in training time, which is usually preferable to increasing the size of a network that is widely deployed to users after the training phase.

Future Work

It may be useful to further explore whether random occlusions as a DA technique is useful outside of specific domains with expected visual occlusions. If so, a comparison of the underlying mathematics to that of drop-out regularization may be in order.

Rotations and random occlusions should be modified to apply mean hue rather than black/gray as they inadvertently modify the training mean image. This change may well improve R360 and RO(n) performance significantly, as performance of CNNs is sensitive to normalization. Wang used a similar zero-pad fill technique in [48] and found no improvement with rotations. Further, a lesser extent of rotation may provide better results than unconstrained rotation as it may more closely match the operational domain of aircraft refueling.

Lastly, recognizing an approaching aircraft is not the only problem yet to be solved in the AAR domain. The problem of estimating the range and relative orientation of an approaching aircraft purely from visual cues is still open. We suspect that com-

puter graphics techniques could be used to generate synthetic training imagery from the refueling perspective of an AAR tanker. Conceivably, the relative position and orientation of an approaching aircraft could be estimated using regression techniques.

IV Scholarly Article: Toward Ensemble Compression with Confidence

Robert Mash, John Pecarina PhD, Brett Borghetti PhD,

Brian Woolley PhD, Scott Nykl PhD

Abstract

Individual neural networks are notorious for claiming high confidence in incorrect outputs. However, ensemble structures naturally provide useful confidence estimates through plurality voting systems or other methods. Ensemble confidence has been shown to be correlated with correct outputs, a useful quality for automated decision systems. This thesis shows that a modification to Bucilua style ensemble compression can *teach* a *student* network an approximation of an ensemble's classification function, as well as a confidence estimating function. Furthermore, the student network's confidence estimate, like the ensemble confidence estimate, is shown to be correlated with the correctness of its outputs. In this way, we improve the decision making usefulness of a student network while adding little complexity.

Introduction

Increasingly, autonomous agents are being deployed into domains that require an elevated level of integrity and reliability. Such domains include tasks in the medical [57] field, municipal law enforcement[58], and military[59] applications. One particular problem of interest is in aerial refueling[42], wherein an autonomous aerial refueling aircraft must correctly identify the model of an approaching aircraft. In this scenario, acting on incorrect identification information could degrade aircraft tracking and threaten the safety of formation flight. Furthermore, improper aircraft identifi-

cation would result in incorrect fuel transfer settings and may damage the aircraft receiving fuel.

To these classes of problems, Artificial Neural Networks (ANNs) can be applied by the ensemble method[60], which allows for a confidence[61] metric/score to be incorporated into a decision framework[10] for object identification. The plural structure of an ensemble naturally provides a confidence metric[62] derived from a winner-take-all voting system. This confidence term is correlated with the correctness of the ensemble's output and is therefore useful for making good decisions[63]. Additionally, weak learner theory tells us that ensembles of neural networks typically outperform individuals in terms of overall classification or regression accuracy[64]. At the same time, as ensembles tend to be computationally complex, some approaches have been developed to decrease their size while retaining their performance. For example, in [13] and [65] Bucilua and Hinton show that ensembles of neural networks can be compressed into a single smaller network, with only slight performance loss.

Some have attempted to enhance compression with meta-information extracted from the ensemble. Specifically, Hu et al. in [66] provided a general framework for mutual distillation, but their procedure specifies an iterative back-and-forth training process, regularizing a Deep Neural Network (DNN) with a structured knowledge source. In [67], Korattikara et al. modified the distillation techniques advanced in [65] with the primary goal of ensuring that the Probability Density Function (PDF) of the student matches that of the teacher.

Modifications to Bucilua's ensemble compression algorithm can be used to teach a single function to a student network that comprises both the primary classification task as well as a confidence estimate. However, to date, no work has incorporated confidence in this way.

Thus, this work aims to modify Bucilua's compression algorithm and show that

a student network can learn to estimate an ensemble's confidence in addition to the primary task of regression or classification. To this end, this work conducts several modified compression experiments teaching a student network to learn the Simple Plurality, Muhlbaier, and Modified Plurality ensemble confidence measures in addition to the ensemble's primary classification function.

This work claims the following contributions:

1. A modification of Bucilua's compression algorithm to include an ensemble confidence value in addition to the ensemble's primary output. Specifically, the label-vector-generation portion is augmented with an ensemble confidence measure.

2. An modified form of the plurality based ensemble confidence metric which aids the confidence learning process.

3. Introduction of the Point Biserial Correlation Coefficient (PBCC) as an appropriate measure of association between teacher or student confidence vs primary task correctness.

The next section discusses the merits of ensemble compression and distillation followed by measures of confidence for individual and ensembles of neural networks. Next, section IV discusses the combination of compression with confidence in the context of a decision framework. While sections IV and IV articulate the experimental methods employed and analyze the experimental results. Section IV discusses the theoretical and practical implications of compression with confidence. Lastly, section IV presents an application example in the domain of automatic aerial refueling.

Related Work

Ensembles.

The performance of Convolutional Neural Networks (CNNs), as well as other types of deep neural networks, can be improved using relatively simple ensemble techniques first explored by Hansen, Michael, and Wolpert in [68], [8], and [69]. More recently in [60], Polikar robustly surveys the field of ensemble decision systems in theoretical and practical terms.

Much like a committee of independent human experts, diverse ensembles of neural networks typically outperform individual neural networks, which is attributed to the decision surface diversity of the ensemble members. Such decision diversity is commonly achieved via training diversity or random initialization of neural network weight parameters. Alternatively, a Random Subspace Ensemble (RSE) is formed by training individual networks with a random subset of available training features as in [9]. In the sub-field of image classification, this practice is accomplished in two ways: a) training individual ensemble members with different images sets, or b) treating the same training set with various forms of data augmentation as in [70].

From an analytical perspective, the decision surface diversity of the ensemble members can be framed as ensemble variance. In [7], Krogh starts with the well known bias-variance trade-off relationship and derives the following representation of ensemble generalization error:

$$E = \bar{E} - \bar{A}, \tag{2}$$

where $\bar{E}$ is the weighed average of the generalization errors in the individual networks and $\bar{A}$ represents the ensemble variance, such that

$$\bar{E} = \sum_{\alpha} w_{\alpha} E^{\alpha} \tag{3}$$

and

$$\bar{A} = \sum_{\alpha} w_{\alpha}(V^{\alpha}(x) - \bar{V}^{\alpha}(x))^2. \tag{4}$$

Note, that α represents individual ensemble member networks and V is the output of an ensemble member.

Equation 2 shows that the generalization error of an ensemble is necessarily smaller than the weighted average generalization error of its ensemble members. Furthermore, equation 2 shows that generalization error is reduced by maximizing ensemble variance. In this way, increasing training diversity results in increased ensemble variance, which in turn results in reduced ensemble generalization error.

Compression.

A common problem limiting the usefulness of large ensembles of deep neural networks is excessive computational complexity. In answer to this issue, Zeng and Martinez in [71], Bucilua in [13], and Hinton in [65] show that a relatively small network can be trained with the goal of reproducing the classification or regression functionality associated with a large ensemble. The effect is to express the performance of a large teacher ensemble into a smaller student network.

Fig. 15 shows the basic ensemble compression algorithm in terms of image classification. Prior to compression, an ensemble of classifiers are individually trained in such a way as to maximize ensemble diversity.

The steps in the following ensemble compression algorithm are repeated until convergence:

1. An ensemble of classifiers individually classify an input image resulting in a 1 x n vector of scalar logits. (The output of ensemble members can be subject to an optional softmax function depending on the flavor of compression/distillation used. The Bucilua based experiments in this work omitted softmax, thus

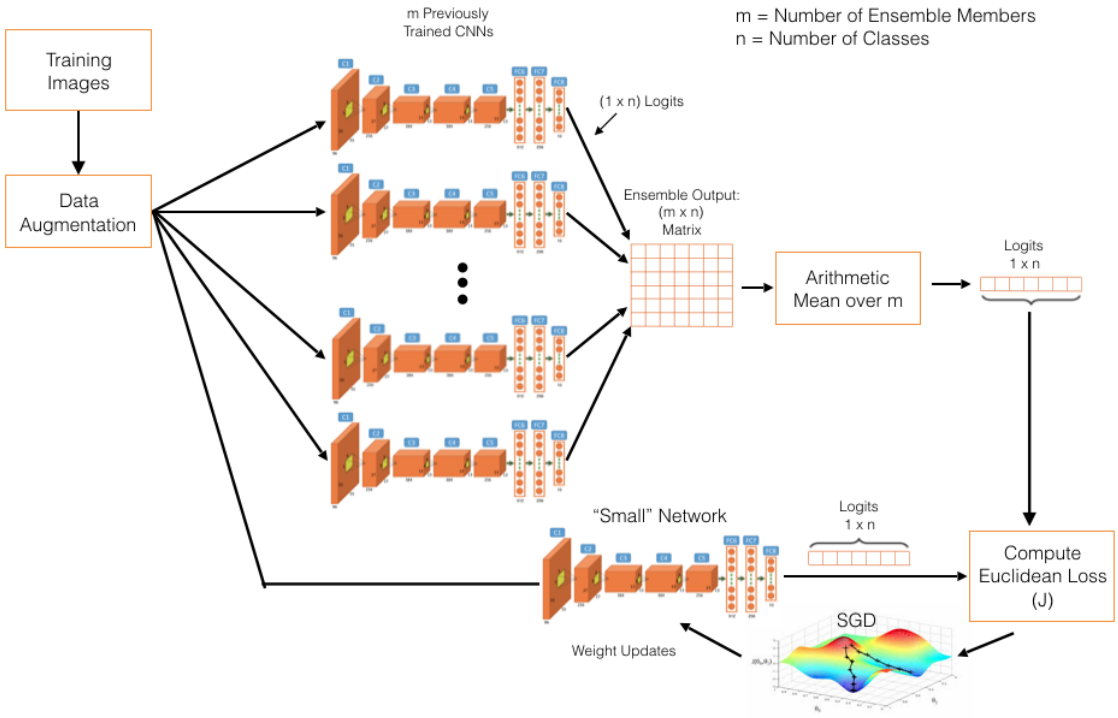


Figure 15. Bucilua et al. ensemble compression. A large teacher network, typically an diverse ensemble of networks, produces a single output vector, which for a given input image, serves as a training label. Stochastic gradient decent is then used to train the small network to perform multi-label regression. In practice, the output of the small network is treated as a classifier, where the maximum valued output is the winning class.

generating logit outputs.)

2. Ensemble member output vectors are assembled into an ensemble output matrix.

3. The ensemble output matrix is then reduced into an output vector via one of several combination[72] methods. The experiments in this work employ the simple arithmetic mean over the ensemble members resulting in a 1 x n vector.

4. The student network is trained via multi-class regression, driven by a Euclidean loss function.

Confidence.

Single Network Confidence.

The deep learning community tends to interpret the softmax output of neural networks as valid Probability Mass Functions (PMFs). However, as established by Duin in [73] and Polikar in [60, Sec. 4] the softmax function is more accurately described as an approximation of the Bayesian posterior probability $P(\omega_1|\mathbf{x})$, of a particular class ω given an input vector $\mathbf{x}$. Duin shows that the softmax function is approximate to posterior probability, but only if a neural network is well trained and sufficiently large. Consequently, in the cases where a neural network is under-sized, or poorly-trained, it can be problematic to refer to a neural network's softmax outputs as confidence values. Particularly, given that the validity as posterior probability, and therefore confidence, depends wholly on how well a neural network is trained to fit the latent function represented by a set of training data. It is for this reason that neural networks are known for occasionally having disappointingly high confidence in an incorrect output. Interestingly, the plural nature of ensembles naturally support a consensus-based voting system that provides a reliable estimate of the correctness of an ensemble's output.

Ensemble Confidence.

Polikar, in [60] examines the merits of Simple Plurality vote based ensemble confidence estimates. Using the Condorcet Jury Theorem[74] Polikar argued that the probability of correctness of the majority vote of an odd number of classifiers with statistically independent outputs, each with prior probability of correctness greater than 50%, must monotonically approach unity when the number of classifiers approaches infinity. Further, Kuncheva in [75, Sec 4.2.1] demonstrated that this is a sufficient, but conservative set of requirements, implying that the Simple Plurality

voting scheme has similar strengths as is consistent with our experimental results presented later.

Muhlbaier et al. in [62] contributes an expanded softmax algorithm that approximates an ensemble's Bayesian posterior probability. Muhlbaier's algorithm takes as input the continuous outputs of each ensemble member and estimates an ensemble confidence value for each possible output class. Notably, Muhlbaier's ensemble confidence is designed to reduce confidence when dissenting votes (i.e., votes for classes other than the class with a plurality of votes) agree, and increase confidence when dissenting votes disagree. This is analogous to a human committee, where the committee's confidence is reduced when there are two competing blocs of opinions, but increased when there is a clear plurality with dissenting opinions spread out amongst possible choices.

Decisions from Confidence.

In non-critical applications, the cost of making an erroneous action as a consequence of incorrect information is slight, therefore it may be reasonable and effective to blindly trust the output of a neural network. However, in critical applications such as aerial refueling, there is great potential for loss of life and assets. In such areas, where the cost of an erroneous decision is great, a more nuanced decision process is warranted.

$$RR = \left(\sum_{N} Accept\&Correct \right) /N \tag{5}$$

$$ReR = \left(\sum_{N} Reject \right) /N \tag{6}$$

$$Reliability = RR + ReR \approx \frac{RR}{\sum_N Accept} \tag{7}$$

In [10], Chow showed that an optimal decision rule given a classifier with a known posterior probablity (i.e., confidence), is to defer action when confidence is below some threshold. Building on Chow's work, Zhang [9] defined a useful nomenclature in equations 5, 6, and 7. Fig. 16 presents an evaluation matrix wherein Recognition Rate (RR) is defined as the probability of a neural network's output being both correct and not rejected (accepted). Rejection Rate (ReR) is the probability of an output being rejected. Reliability is the sum of RR and ReR, or the compliment of the probablity of the decision process yielding an incorrect and erroneously accepted answer.

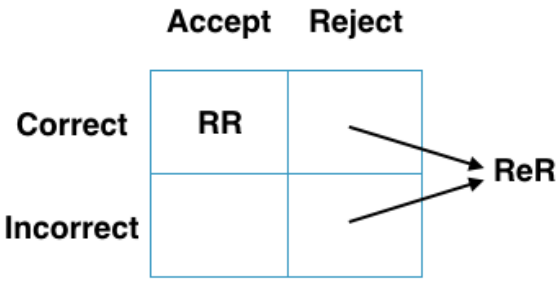


Figure 16. Test set evaluation matrix. This matrix is used to generate reliability curves as in Fig. 22. A rejection threshold is selected, then when evaluating a test set, the matrix is used to organize test instances depending on whether they're correctly classified and if their associated confidence score is higher or lower than the rejection threshold.

Furthermore, Zhang showed that a domain appropriate reliability can be achieved by selecting a trade-off between the Rejection Rate (ReR) and Reliability (Fig. 22). Or put simply, an appropriately high reliability can be selected at the cost of discarding increasing numbers of classifications.

The Need for Compression with Confidence

It is long established in the field of machine learning that ensembles of learners, be they neural networks or otherwise, typically outperform [60] individuals. Also, ensembles are naturally structured to provide a plurality vote based confidence value [62]. As an ensemble's output confidence is correlated with correctness, it can be leveraged to add design flexibility to a decision framework as in [9].

Unfortunately, ensembles often exhibit excessive computationally complexity, in both time and space, which can limit practical application. Typically the computational complexity challenge of ensembles is mitigated by introducing compression, which teaches a student network a function that represents the primary capability of an ensemble. However, as with any individual neural network, it can be problematic to interpret the student network's output activation as posterior probability, or in effect, confidence.

Ideally a compressed student network would learn not only the primary classification or regression capability of an ensemble, but also retain a plurality based confidence estimate which is correlated with correctness. The next section presents a modification to Bucilua's compression algorithm than seeks to realize this ideal.

Methods

This section describes our methods used to compute the simple plurality, Muhlbaier, and modifed plurality ensemble confidence measures from the ensemble's output matrix-of-logits (Fig. 17). And describes the modifications to Bucilua's ensemble compression algorithm used to teach a single function to a student network that comprises both the primary classification task as well as a confidence estimate.

Computing Ensemble Confidence.

Simple Plurality Confidence.

In the first compression experiment ensemble confidence is computed as follows:

1. A vote is generated for each ensemble member by identifying the class index of the maximum valued output logit.
2. A tally is computed by summing the number of votes for each class.
3. Simple Plurality ensemble confidence is the number of votes for the class with a plurality of votes divided by the number of classifier votes (the ensemble size).

Simple Plurality ensemble confidence

$$C(\mathbf{x}) \in \{1/m, 2/m, \cdots, (m-1)/m, m/m\}, \tag{8}$$

where m is the ensemble size and $\mathbf{x}$ is the input vector.

Muhlbiaer's Ensemble Confidence.

In the second experiment ensemble confidence for class j is computed with Muhlbaier's expanded softmax method as in [62, Eqns 3 & 4], reproduced (with adjustments) here as

$$C_j(\mathbf{x}) = \frac{e^{F_j(\mathbf{x})}}{\sum_{k=1}^{N} e^{F_k(\mathbf{x})}} \tag{9}$$

and

$$F_j(\mathbf{x}) = \sum_{t=1}^{N} \begin{cases} logit_{t,j} & h_t(\mathbf{x}) = \omega_j \\ 0 & otherwise \end{cases} \tag{10}$$

where j is class index, N is number of classes, h is the ensemble's class hypothesis, ω_j is class j, and $logit_{t,j}$ is an entry at location (t, j) in the ensemble's output matrix of

logits. In order to generalize these equations, the Learn++ specific normalized error term $log(\frac{1}{\beta_t})$ from [62, eqn 4] is replaced with an indexed ensemble output logit.

In experiment two, confidence is only computed for the hypothesized class j as determined by taking the arithmetic mean over the ensemble's output matrix and finding the maximal value. Muhlbaier's ensemble confidence resides on the interval [0,1] and is largest when the ensemble member's are unamimous in their *votes*. The F_j function in the denominator tends to increase confidence when dissenting votes are spread-out and reduce confidence when dissenting votes are unanimous.

Modified Plurality Ensemble Confidence.

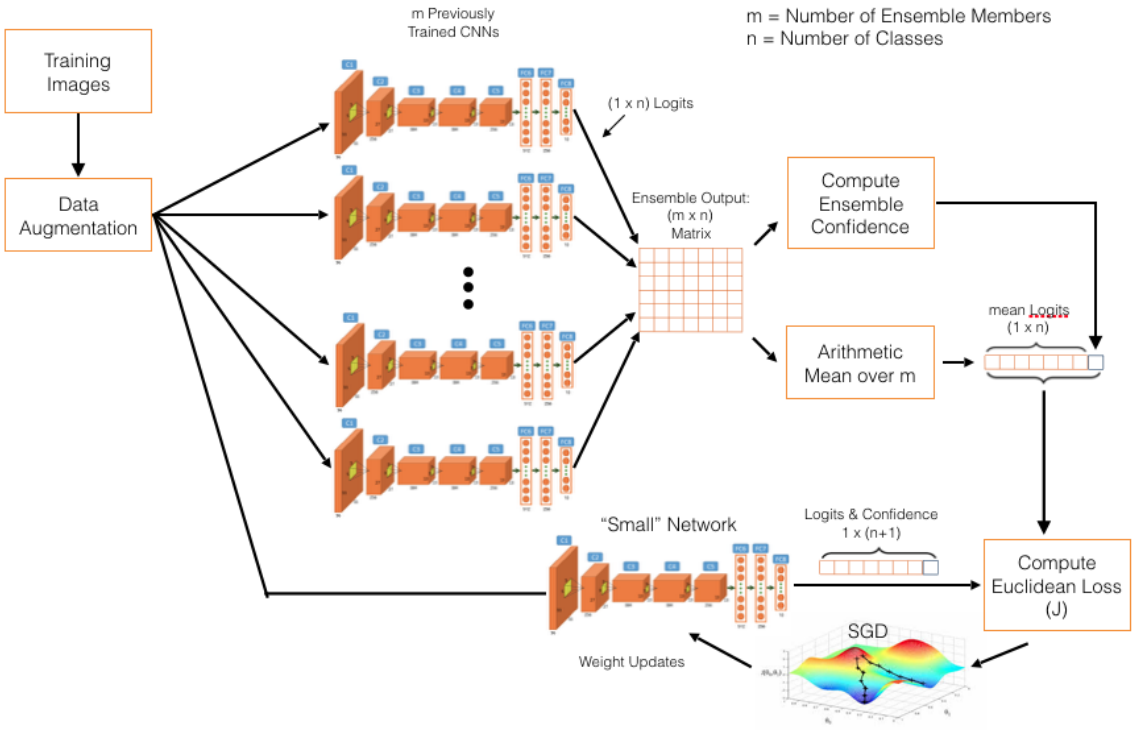


Figure 17. Modified Bucilua et al. ensemble compression [13]. Using the output matrix of the large teacher network, an ensemble confidence value is computed using any of several methods. This value is then concatenated with the ensemble's mean combined output vector, which for a given input image, serves as a training label. Stochastic gradient decent is then used to train the small network to perform multi-label regression. The student network then learns to classify imagery as well as estimate the confidence of the teacher ensemble.

Experiment three computes ensemble confidence with Modified Plurality, a modified form of Simple Plurality with a confidence reduction term similar to Muhlbaier's method.

Modified Plurality ensemble confidence is computed as follows:

1. A vote is generated for each ensemble member by identifying the class index of the maximum valued output logit.
2. A vote tally is computed by summing the number of votes for each class.
3. P1 and P2 are defined as the number of votes for the class with voting plurality and the number of votes for the runner-up, respectively.
4. Modified Plurality ensemble confidence $= \frac{P1-P2}{m}$, where m is the ensemble size.

Modified Plurality ensemble confidence

$$C(\mathbf{x}) \in \{0/m, 1/m, \cdots, (m-2)/m, m/m\}, \tag{11}$$

where m is the ensemble size. Note that $(m-1)/m$ is specifically excluded. In this way the Simple Plurality is modified to reduce confidence when dissenting votes are more unanimous. The effect of the dissent term is to skew confidence downward unless the ensemble voting is unanimous.

Modified Compression Algorithm.

The experiments performed in this work utilized a modified form of ensemble compression (Fig. 17). There are primarily two differences from the basic ensemble compression training algorithm:

1. Ensemble confidence is computed from the ensemble output logit matrix.

2. The resultant ensemble confidence scalar is concatenated with the (1 x n) mean logits vector. This is the new target label that corresponds to the input training image.

Experiments one, two, and three utilize the Simple Plurality, Muhlbaier's method, and Modified Plurality ensemble confidence techniques respectively.

In these experiments, the ensemble is composed of identically sized Convolutional Neural Networks (CNNs) as described in [**?**]. Each CNN is trained individually with the same ten-class training dataset of 60,000 images but with varying types of data augmentation as described in [70]. The student network is a variant of the Alexnet CNN from [12] with the final fully connected layer configured to have eleven outputs: Ten outputs for the ten classes of images and a single output for the ensemble confidence estimate.

The training dataset from [42] and [70] is used as a transfer set as discussed in [13], and [65]. Training of the student network occurs over 250 training epochs with the following parameters:

1. Batchsize = 150
2. Optimization by stochastic gradient descent with momentum
3. Fixed learning rate = 3e-4
4. Momentum = 0.9
5. Weight decay = 0.0005

Evaluation of classification accuracy of the student network is performed with the same 15,000 member test set used in [42] and [70]. After each forward pass, the student network's Logits & Confidence output vector is recorded for analysis as presented in the next section.

Compression with Confidence: Results

Histograms can be used to approximate the distributions of ensemble confidence for cases of correct and incorrect ensemble classifier output. In the correct case, confidence is typically approximate to unity, while in the incorrect case, confidence is much lower. The normalized histogram in Fig. 18 (top)(left) illustrates in green the distribution of the Simple Plurality ensemble confidence when the ensemble correctly classifies a test image. Similarly, the normalized distribution of Simple Plurality ensemble confidence with incorrect ensemble classification is shown in red.

Table 2. Ensemble confidence vs learned confidence. Confidence calculated by (top) simple Plurality, (middle) Muhlbaier's method, and (bottom) Modified Plurality.

Confidence Type	**Confidence to Correctness Correlation Coefficient**	**Mean Correct Confidence**	**Mean Incorrect Confidence**
Ensemble Plurality	0.520	0.941	0.512
Learned Plurality	0.472	0.943	0.685
Ensemble Muhlbaier	0.617	0.983	0.620
Learned Muhlbaier	0.476	0.993	0.859
Ensemble Modified Plurality	0.501	0.902	0.257
Learned Modified Plurality	0.487	0.926	0.520

The box plot in Fig. 18(bottom)(left) characterizes the same correct and incorrect distributions continuously, illustrating the median value in red, the interquartile range as the box, and the extent of the outlier thresholds at 1.5 interquartile ranges outside the 2nd and 3rd quartiles. Individual outliers are plotted outside the outlier range whiskers.

In order to compare the confidence estimating capability of a trained student network, similar normalized histograms and box plots are presented in Fig. 18 (top) and (bottom) respectively. As illustrated in Fig. 18, the regressed confidence measure demonstrates behavior similar to the computed simple plurality ensemble confidence

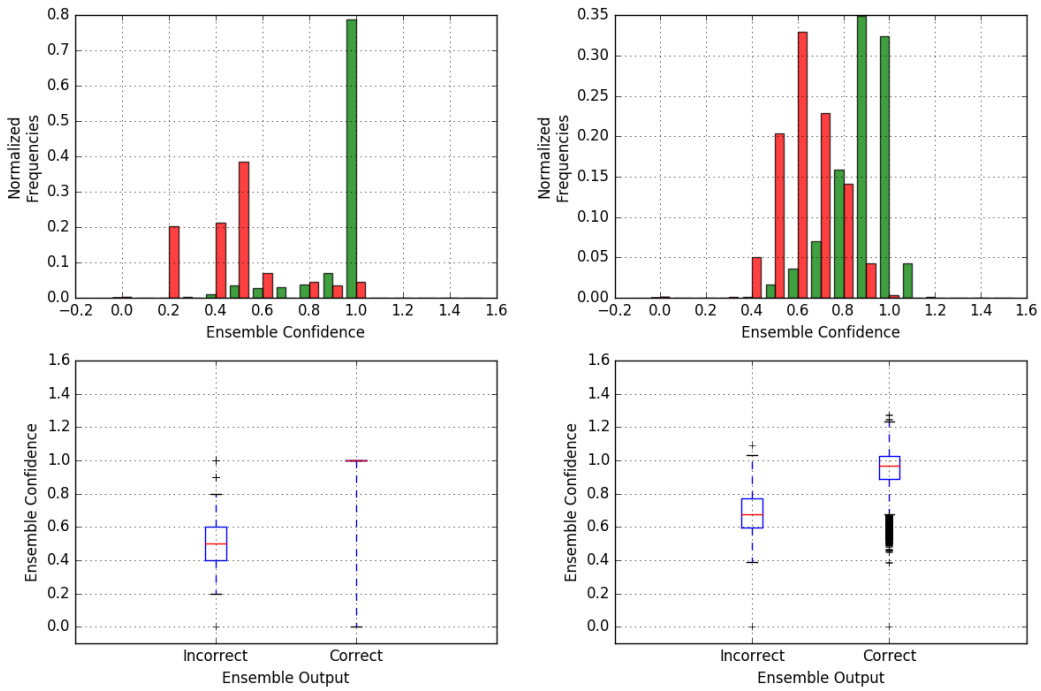


Figure 18. (Left) Ensemble Confidence computed from simple Plurality. (Right) Student network's learned confidence estimate.

measure depending on whether the student's classification output is correct. Note, that as the student network regresses a confidence measure, the range of confidence values is not constrained to the $[0, 1]$ interval.

Identical to Fig. 18 with Simple Plurality confidence, Fig. 19 and Fig. 20 correspond to the Muhlbaier and Modified Plurality ensemble confidence measures respectively. The student networks in these cases demonstrate regressed confidence which is correlated with the correctness of the student's classification output.

In all experiments, as shown in Table 2, the regressed confidence output of the student network is positively correlated with its classification correctness to a similar degree as that of the teacher ensemble. The implication of this finding is that the regressed confidence measures are of similar utility as the computed ensemble confidence measures. Further, there is a reduction in the magnitude of the confidence-to-correctness correlation coefficient when comparing an ensemble confidence measure to

a student regressed confidence value. Of the ensemble confidence measures evaluated, the modified plurality technique results in the most similar ensemble vs regressed correlation coefficients.

As shown in Table 2, all three ensemble confidence measures result in similar regressed correlation coefficients. Consequently, it is not expressly clear which ensemble measure is most suitable for training a student to regress a useful confidence measure. However, these results do show that the utility of ensemble confidence measures in terms of correlation to classification-correctness is largely preserved in student networks.

Table 3. Classifier Performance: Test set classification accuracy (%) after training to convergence (250 training epochs).

Ensemble	Compressed without Confidence	Compressed with Confidence (plurality)	Compressed with Confidence (Muhlbaier)	Compressed with Confidence (Modified Plurality)
95.8	93.0	93.1	93.1	93.1

Discussion

The experiments conducted were aimed at teaching student networks the Simple Plurality, Muhlbaier, and Modified Plurality ensemble confidence values whilst also approximating the primary image classification function embodied in the ensemble. Here we discuss what ensemble confidence is and how it's learned in the context of training deep neural networks.

The strength of deep neural networks is their ability to learn a hierarchy of abstract features. This is accomplished by framing the entire learning process as a large optimization problem, solvable via gradient decent. As discussed in [60], and [9], confidence can also be thought of as a distance metric from a decision surface in

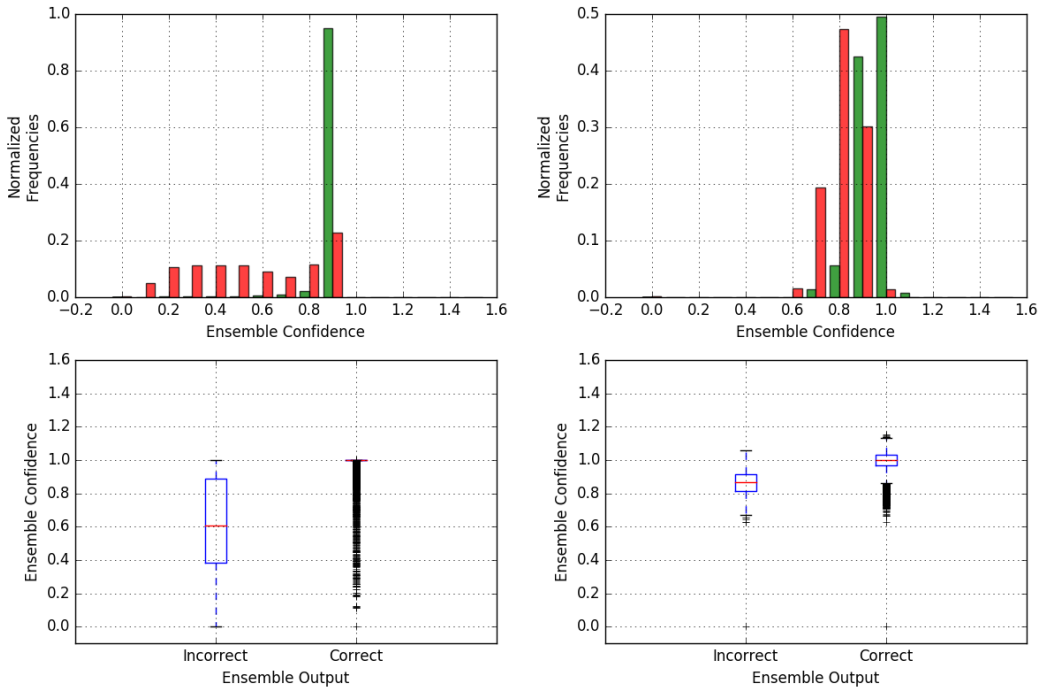


Figure 19. (Left) Ensemble Confidence computed with Muhlbaier's method. (Right) Student network's learned confidence estimate.

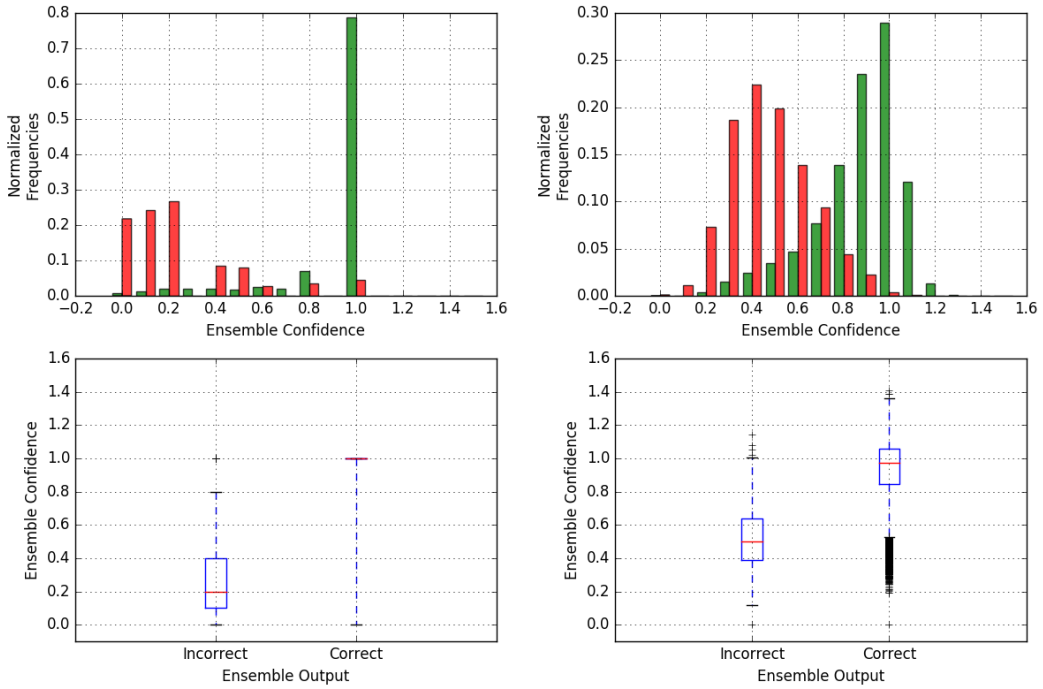


Figure 20. (Left) Ensemble Confidence computed with Modified Plurality. (Right) Student network's learned confidence estimate.

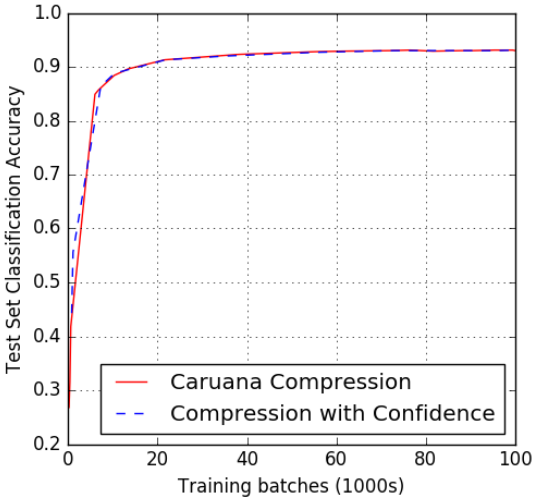


Figure 21. Ensemble learning curves. The addition of confidence to the target label doesn't appreciably show down the learning process.

the parameter space realized by the neural network weights. When viewed from this perspective, confidence is then a descriptive measure of the self-performance of a neural network. In the experiments presented here, it is clear that ensemble confidence is not perfectly reproduced. However, as the experimental results show, a modified form of ensemble confidence has been learned, therefore it stands to reason that ensemble confidence can itself be represented as an abstraction of hierarchically related features.

We can establish that performance of the student neural network is largely unaffected by the presence of the additional confidence output. As detailed in Table 3, the test set classification accuracy of the ensemble, the compressed network without confidence, and the compressed network with Plurality, Muhlbaier, and Modified Plurality confidence measures. Each student network resulting from compression has essentially identical test set classification performance. This observation carries the implication that either learning to approximate an ensemble confidence measure doesn't require much capacity relative to the size of the network, or that the network is over sized. Furthermore, as shown in Fig 21, the addition of confidence to the tar-

get label doesn't appreciably slow down the learning process. The two observations in combination strongly suggest that the addition of confidence does not significantly increase the required capacity relative to the primary classification task.

Application Example

The domain of Automatic Aerial Refueling (AAR) is critical in nature as any mid-air mishap is potentially catastrophic to either or both of the tanker aircraft, or a fuel-receiving, or receiver aircraft. Therefore, AAR is a reasonable domain in which to make decisions informed with confidence. AAR is not performed without previous mission planning, or without constant radio communication. However, even with a plan and with communications available, it is prudent to visually identify the model of an approaching receiver aircraft. To this end, manned tanker aircraft utilize personnel to visually identify the model of receiver aircraft on refueling approach. This work is part of a larger body of work the purpose of which is to build and test an automated visual aircraft identification system. Prior to, and in support of this work, an ensemble of identical Convolutional Neural Networks (CNNs) were trained and evaluated using an internet-sourced body of imagery roughly evenly representing ten models of U.S. Air Force aircraft. The dataset totals 76,000 images divided into separate training and test sets with 80% / 20% proportion. Each member of the CNN ensemble was separately trained using the same training dataset. A range of image transformations corresponding to plausible physical transformations was selected. In this way, a different type of training data augmentation was used for training each ensemble member. This work is detailed in [42], and [70].

For the AAR application example pictured in Fig. 22, we select the student network from experiment three which learned to estimate the Modified Plurality form of ensemble confidence.

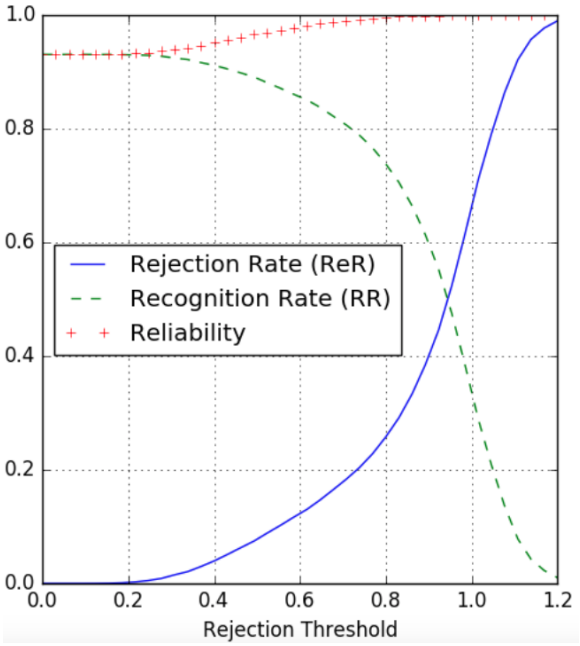


Figure 22. Aerial refueling application example. Using the Chow[10] reject option decision framework, image recognition reliability can become a design parameter. Selection of a high reject threshold selects high system reliability by discarding a higher proportion of images where the confidence is less than the rejection threshold.

We create a range of rejection threshold values representing the interval [0,1.2]. For each rejection threshold we:

1. Evaluate the aircraft image test dataset.

2. Populate the evaluation matrix in Fig. 16 according to whether the student correctly classified the image, and whether the student's confidence output was above the rejection threshold (accepted) or below (rejected).

3. Evaluate RR, ReR, and Reliability from equations 5, 6, and 7.

The resulting data, plotted in Fig. 22 then represents a design trade-off space. The recognition rate RR is the probability that an image is correctly classified and that the confidence for that image is higher than the rejection threshold. This is the upper-left region in the matrix in Fig. 16. The rejection rate ReR is simply the probability that the student's confidence output is below the rejection threshold.

This is the right half of the matrix in Fig. 16. Both the RR and ReR regions in the matrix in Fig. 16 represent reliable operation of the decision system where either the confidence is too low on which to act, or the confidence is high and the image is correctly classified.

Chow, in [10], showed that we can reduce the probability represented by the lower-left quadrant of the matrix in Fig. 16 to arbitrarily low values by increasing the rejection threshold. The trade-off is that increasing the rejection threshold to increase reliability discards an increasingly large proportion of image classifications.

In the AAR example with a critically high cost of failure, one might select a rejection threshold that corresponds to a high reliability and accept a higher rejection rate. Additionally, in an operational setting, if too many images are discarded due to low confidence, the problem would be elevated to an appropriate decision authority in the command and control structure.

Conclusions

In the field of autonomy, there is a class of critical, high reliability problems to which ensembles of neural networks can appropriately be applied. The benefits of which are generally higher performance, and a natural, voting based confidence system which opens up the possibility of using a high reliability, rejection-option based decision framework. Unfortunately, ensembles are sometimes ungainly and complex which limits their employment in size-weight-power constrained environments. Bucilua[13], et al. have addressed this issue with ensemble compression, a technique that teaches a smaller student network to approximate the function expressed by the ensemble. It should be noted however, that such an approach removes the voting-confidence benefit provided by an ensemble.

We have shown through three experiments that the compression algorithm popu-

larized by Bucilua et al, can be modified to teach a student network not only to learn the teacher ensemble's primary function, but to also learn how to estimate what the teacher ensemble's voting-based confidence would be for images not contained in the training set.

We claim the following contributions:

1. A modification of Bucilua's compression algorithm to include an ensemble confidence value in addition to the ensemble's primary output. Specifically, the label-vector-generation portion is augmented with an ensemble confidence measure.

2. A modified form of the plurality based ensemble confidence metric which aids the confidence learning process.

3. Introduction of the Point Biserial Correlation Coefficient (PBCC) as an appropriate measure of association between teacher or student confidence vs primary task correctness.

V Conclusions & Recommendations

Conclusions

This chapter considers the individual conclusions from the scholarly works presented in chapters II, III, and IV in the context of the over-arching and subsidiary research questions presented in chapter one.

The over-arching investigative question was: 'Can the current state of the art in object classification successfully identify aircraft on approach as reliably as a member of trained aircrew?'

Chapter I hypothesized that a convolutional neural network, sufficiently and properly trained with appropriate training imagery, would succeed in this task in much the same way that a member of aircrew learns to identify aircraft visually given enough prior visual examples. Investigation of this over-arching research question was broken up into three sequentially related areas of inquiry as follows.

1. What set of network design hyper-parameters, which describe the network shape and capacity, optimize object classification performance in the specific domain of aircraft recognition?

   The work in chapter II utilized the fundamentals of machine learning to arrive at an approximately optimal compromise between bias and variance test errors. The final set of network hyper-parameters achieved 91.5% test accuracy, representing a test error minimum on the bias-variance trade-off curve. In this way, an approximately optimal set of hyper-parameters were developed for the domain of aircraft recognition.

2. What combination of data augmentation techniques is most applicable to the domain specific task of classifying aircraft?

   The work in chapter III evaluated several plausible physical transformations that could occur in the aircraft recognition domain. Image transformations which mimic these physically plausible transformations were developed and used to augment the size of the training dataset. The resulting trained instances of AfCaffe yielded improvements in performance above the version of AfCaffe arrived at in Chapter II. Circumstance required early stopping of the training of the AfCaffe instances reported in chapter III. However, continued training completed after publication of the work reported in chapter III, resulted in individual AfCaffe networks with 93.5% test accuracy. Therefore the data augmentation techniques explored in chapter III, which are specific to the aerial refueling domain, conferred 45.7% reduction in test error over the results presented in chapter II.

3. Can existing techniques in ensemble combination and compression be modified to yield a computationally manageable network while preserving both the favorable performance of a large ensemble as well as a robust confidence measure?

   The work represented in chapter IV, has three important facets. First, an ensemble was formed from the diverse AfCaffe instances individually trained in chapter III. The test performance of that ensemble rose to 95.8% test set accuracy, another substantial improvement in classification performance. Second, in order to retain 93.1% test set accuracy, the size of the 'student' network was expanded up to the original Alexnet configuration as represented in line 8 of Table 1 in Chapter II. The number of neurons in Alexnet is smaller than that of the AfCaffe ensemble, but substantially larger than that of a single instance of AfCaffe. In terms of performance, this result, while consistent with Caruana's

compression results, could likely be improved by applying Hinton's distillation technique. Therefore, while not completed in this work, it's likely that the computational complexity of the AfCaffe ensemble could be reduced further in future work by adapting Hinton's distillation technique.

Third, the modification to Caruana's compression technique in chapter IV did result in a small 'student' network with a confidence output. This work included an application example detailing a reject-option decision framework which utilizes the small network's confidence output. The effect is to increase the reliability of the network's output by trading off the proportion of classification events with insufficient confidence to be trusted.

The over-arching research question: 'Can the current state of the art in object classification successfully identify aircraft on approach as reliably as a member of trained aircrew?', remains strictly unanswered. However, future work applying the neural network sizing, training, and confidence-augmented-compression techniques outlined in these scholarly works, in conjunction with future refueling-perspective datasets is likely to provide a conclusive answer.

Recommendations

Here follow several recommended areas of additional inquiry:

1. The hyperparameter optimization work presented in chapter II was constrained to the fully connected classifier portion of Alexnet. This experiment was structured to reduce the parameter count, and thus the computational complexity of Alexnet. Similar work could seek to optimize test set classification performance in the space of hyperparameters that define the structure of the convolutional portion of Alexnet.

2. The dataset presented in chapter II is composed of thousands of images gathered from the internet from all manner of popular photographic perspectives. However, these techniques should be applied to a future dataset composed of solely refueling perspective aerial photography.

3. There are a class of physically plausible transformations in the aircraft recognition domain, such as subject rotation and translation, that are mimicable with simple image transformations. Therefore, the work in chapter III could be extended by utilizing a suite of computer graphics to create synthetic imagery on demand for training a CNN.

4. The field of deep learning has progressed beyond Alexnet, which was introduced in 2012. Therefore, the techniques explored in this thesis should be evaluated over more recently developed networks such as GoogleNet, VGG-Net, and ResNet.

5. In order to evaluate the relative performance of a CNN-based aircraft classifier against that of humans, an appropriate performance metric should be developed to quantize human aircraft recognition.

Bibliography

1. Paul Norwood and Benjamin Jensen, "Wargaming the Third Offset Strategy," *Joint Force Quarterly*, vol. 4, pp. 34–39, 2016.

2. Pierre Baldi and Peter Sadowski, "Understanding Dropout," in *Neural Information Processing Systems Conference*, 2012, pp. 1–9.

3. Nathaniel Sauder, "Encoded Invariance in Convolutional Neural Networks," *University of Chicago*, pp. 2–6, 2006.

4. Yann LeCun, Yashua Bengio, and Geoffrey Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

5. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

6. Angjoo Kanazawa, Abhishek Sharma, and David Jacobs, "Locally Scale-invariant Convolutional Neural Network," *Deep Learning and Representation Learning Workshop: Neural Information Processing Systems Conference*, pp. 1–11, 2014.

7. Anders Krogh and Jesper Vedelsby, "Neural Network Ensembles, Cross Validation, and Active Learning," *Advances in Neural Information Processing Systems 7*, pp. 231 – 238, 1995.

8. Perrone Michael and Leon Cooper, "When Networks Disagree: Ensemble Methods for Hybrid Neural Networks," Tech. Rep., 1992.

9. Bailing Zhang, "Random subspace support vector machine ensemble for reliable face recognition Bailing Zhang," *International Journal of Biometrics*, vol. 6, no. 1, 2014.

10. C. K. Chow, "On Optimum Recognition Error and Reject Tradeoff," *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 41–46, 1970.

11. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Convolutional Architecture Feature Embedding," in *Association for Computing Machinery Multimedia Open Source Software Competition*, 2014.

12. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances In Neural Information Processing Systems*, pp. 1–9, 2012.

13. Cristian Bucilu, Rich Caruana, and Alexandru Niculescu-Mizil, "Model compression," *Proceedings of the 12th Association for Computing Machinery - Special Interest Group on Knowledge Discovery and Data Mining - international conference on Knowledge discovery and data mining*, p. 535, 2006.

14. Sander Dieleman, Kyle Willett, and Joni Dambre, "Rotation-invariant convolutional neural networks for galaxy morphology prediction," *Monthly Notices of the Royal Astronomical Society*, vol. 450, no. 2, pp. 1441–1459, 2015.

15. Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Man, Rajat Monga, Sherry Moore, Derek Murray, Jon Shlens, Benoit Steiner, Ilya

Sutskever, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Oriol Vinyals, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *Google Research*, p. 19, 2015.

16. Yoshua Bengio, "Practical recommendations for gradient-based training of deep architectures," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7700, pp. 437–478, 2012.

17. David Hubel and Torsten Wiesel, "Receptive Fields and Functional Architecture of Monkey Striate Cortex.," *The Journal of physiology*, vol. 195, no. 1, pp. 215–43, 1968.

18. Warren Mcculloch and Walter Pitts, "A logical calculus nervous activity," *Bulletin of Mathematical Biology*, vol. 52, no. l, pp. 99–115, 1990.

19. Paul Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph.D. thesis, Harvard University, 1974.

20. Yann LeCun, Leon Bottou, Yashua Bengio, and Patrick Haffner, "Gradient Based Learning Applied to Document Recognition," *Proc fo the IEEE*, vol. 1, no. November, pp. 1–46, 1998.

21. Kunihiko Fukushima, "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

22. David Rumelhart, Geoffrey Hinton, and Ronald Williams, "Learning Internal Representations by Error Propagation," *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence*, pp. 399–421, 2013.

23. Yann LeCun, Bernard Boser, John. Denker, and Et Al., "Handwritten Digit Recognition with a Back-Propagation Network," *Advances in Neural Information Processing Systems*, pp. 396–404, 1990.

24. Yann LeCun, Bernard Boser, John. Denker, and Et Al., "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

25. Yann LeCun, Leon Bottou, and Yoshua Bengio, "Reading checks with Multilayer Graph Transformer Networks," in *IEEE Proc. of Computer Vision and Pattern Recognition*. Speech and Image Processing Services Research Lab, 1997, pp. 0–3.

26. Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio, "Robust Object Recognition with Cortex-Like Mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 3, March 2007*, vol. 29, no. 3, pp. 411–426, 2007.

27. Christian Szegedy, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

28. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *The IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

29. Pierre Sermanet, Soumith Chintala, and Yann LeCun, "Convolutional neural networks applied to house numbers digit classification," *Proceedings of International Conference on Pattern Recognition*, pp. 10–13, 2012.

30. Fan Hu, Gui-Song Xia, Jingwen Hu, and Liangpei Zhang, "Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14680–14707, 11 2015.

31. Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun, "What is the best multi-stage architecture for object recognition?," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2146–2153, 2009.

32. Yoshua Bengio and Yann LeCun, "Scaling Learning Algorithms towards AI," *Large Scale Kernel Machines*, , no. 1, pp. 321–360, 2007.

33. Xiaojie Jin, Chunyan Xu, Jiashi Feng, Yunchao Wei, Junjun Xiong, and Shuicheng Yan, "Deep Learning with S-shaped Rectified Linear Activation Units," *Accepted by the Association for the Advancement of Artifiicial Intelligence (AAAI)*, 2015.

34. George Dahl, Tara Sainath, and Geoffrey Hinton, "Improving Deep Neural Netowrks for Larfe Vocabulary Continuous Speech Recognition (LVCSR) Using Recitified Linear Units and Dropout, Department of Computer Science , University of Toronto," *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013.

35. Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-Rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

36. Dumitru Erhan, Aaron Courville, and Pascal Vincent, "Why Does Unsupervised Pre-training Help Deep Learning ?," *Journal of Machine Learning Research*, vol. 11, no. 2007, pp. 625–660, 2010.

37. Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Return of the Devil in the Details : Delving Deep into Convolutional Nets," *British Machine Vision Conference,*, pp. 1–11, 2014.

38. Maxime Oquab and Leon Bottou, "Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks," *Conference on Computer Vision and Pattern Recognition*, 2014.

39. Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus, "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks," *Advances in Neural Information Processing Systems 28*, pp. 1486–1494, 2015.

40. Soheil Bahrampour, Naveen Ramakrishnan, Lukas Schott, and Mohak Shah, "Comparative Study of Caffe, Neon, Theano, and Torch for Deep Learning," *Knowledge Discovery and Data mining (KDD)*, 2016.

41. Steven M Ross, "Formation Flight Control for Aerial Refueling," *Master's Thesis*, p. 183, 2006.

42. Robert Mash, Nicholas Becherer, Brian PhD Woolley, and John PhD Pecarina, "Toward Aircraft Recognition With Convolutional Neural Networks," in *Proceedings of the IEEE National Avionics & Electronics Conference*, 2016.

43. Francesco Visin, Kyle Kastner, Kyunghyun Cho, Matteo Matteucci, Aaron Courville, and Yoshua Bengio, "ReNet: A Recurrent Neural Network Based Alternative to Convolutional Networks," *arXiv:1505.00393*, 2015.

44. Andrej Karpathy, George Toderici, Sachin Shetty, Tommy Leung, Rahul Sukthankar, and Li Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1725–1732.

45. Karen Simonyan and Andrew Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," *arXiv preprint arXiv:1406.2199*, pp. 1–11, 2014.

46. Ozgur Yilmaz, "Classification of Occluded Objects using Fast Recurrent Processing," *arXiv:1505.01350*, 2015.

47. Refik Can Malli, Mehmet Aygun, and Hazim Kemal Ekenel, "Apparent Age Estimation Using Ensemble of Deep Learning Models," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

48. Keven Kedao Wang, "Image Classification with Pyramid Representation and Rotated Data Augmentation on Torch 7," *Standford CS231n Course Project Reports*, 2015.

49. Bojan Pepikbo, Rodrigo Benenson, Tobias Ritschel, and Bernt Schiele, "What is holding back convnets for detection?," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9358, pp. 517–528, 2015.

50. Konstantinos Rematas, Tobias Ritschel, Mario Fritz, and Tinne Tuytelaars, "Image-based synthesis and re-synthesis of viewpoints guided by 3D models," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3898–3905, 2014.

51. Michael Stark, Michael Goesele, and Bernt Schiele, "Back to the Future: Learning Shape Models from 3D CAD Data," *British Machine Vision Conference,*, pp. 1–11, 2010.

52. Jiaolong Xu, David Vazquez, Antonio M. Lopez, Javier Marin, and Daniel Ponsa, "Learning a multiview part-based model in virtual world for pedestrian detection," *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 467–472, 2013.

53. Scott Nykl, Chad Mourning, and David Chelberg, "Interactive mesostructures with volumetric collisions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 7, pp. 970–982, 2014.

54. Ian Goodfellow, Jean Pouget-Abadie, and Mehdi Mirza, "Generative Adversarial Networks," *Advances in Neural Information Processing Systems 27*, 2014.

55. Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *arXiv:1511.06434*, pp. 1–15, 2015.

56. Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox, "Learning To Generate Chairs With Convolutional Neural Networks," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1538–1546, 2015.

57. Yungang Zhang, Bailing Zhang, Frans Coenen, and Wenjin Lu, "Highly Reliable Breast Cancer Diagnosis with Cascaded Ensemble Classifiers," *IEEE World Congress on Computational Intelligence*, pp. 10–15, 2012.

58. Giles Oatley, Brian Ewart, and John Zeleznikow, "Decision support systems for police: lessons from the application of data mining techniques to "soft" forensic evidence," *Artificial Intelligence Law*, vol. 14, no. 1, pp. 35–100, 2006.

59. A. Khashman, "Automatic Detection of Military Targets utilising Neural Networks and Scale Space Analysis," Tech. Rep., 2000.

60. Rich Polikar, "Ensemble based systems in decision making," *Circuits and Systems Magazine, IEEE*, vol. 6, no. 3, pp. 21–45, 2006.

61. Leijun Li, Qinghua Hu, Xiangqian Wu, and Daren Yu, "Exploration of classification confidence in ensemble learning," *Pattern Recognition*, vol. 47, no. 9, pp. 3120–3131, 2014.

62. Michael Muhlbaier and Apostolos Topalis, "Ensemble confidence estimates posterior probability," *Multiple Classifier Systems*, pp. 326–335, 2005.

63. Kristine Monteith and Tony Martinez, "Using multiple measures to predict confidence in instance classification," *Proceedings of the International Joint Conference on Neural Networks*, 2010.

64. Robert E Schapire, "The Strength of Weak Learnability (Extended Abstract)," *Machine learning*, vol. 227, pp. 28–33, 1989.

65. Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the Knowledge in a Neural Network," *Neural Information Processing Systems Conference 2014 Deep Learning Workshop*.

66. Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric P Xing, "Deep Neural Networks with Massive Learned Knowledge," *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16)*, pp. 1670–1679, 2016.

67. Anoop Korattikara, Vivek Rathod, Kevin Murphy, and Max Welling, "Bayesian Dark Knowledge," *Advances in Neural Information Processing Systems 28*, 2015.

68. Lars Kai Hansen and Peter Salamon, "Neural Network Ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993, 1990.

69. David H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

70. Robert Mash, Brett Borghetti, and John Pecarina, "Improved Aircraft Recognition for Aerial Refueling through Data Augmentation in Convolutional Neural Networks," *Proceedings of the IEEE National Avionics & Electronics Conference*, 2016.

71. X. Zeng and T. R. Martinez, "Using a neural network to approximate an ensemble of classifiers," *Neural Processing Letters*, vol. 12, no. 3, pp. 225–237, 2000.

72. Varuna Tyagi, "A Survey on Ensemble Combination Schemes of Neural Network," *International Journal on Computer Applications*, vol. 95, no. 16, pp. 18–21, 2014.

73. Robert Duin and David Tax, "Classifier Conditional Posterior Probabilities," *Advances in Pattern Recognition*, vol. 1451, pp. 611–619, 2005.

74. Lloyd Shapley and Bernard Grofman, "Optimizing group judgmental accuracy in the presence of interdependencies," *Public Choice*, vol. 43, no. 3, pp. 329–343, 1984.

75. Ludmila I. Kuncheva, "Combining Pattern Classifiers," *Methods and Algorithms*, pp. 45–94, 2004.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
23–03-2017	Master's Thesis	Sep 2015 — Mar 2017

4. TITLE AND SUBTITLE

Toward Automated Aerial Refueling: Automated Visual Aircraft Identification with Convolutional Neural Networks

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Mash, Robert L. Captain, USAF

5d. PROJECT NUMBER

16G189

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-17-M-048

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Ba T. Nguyen
DR-03, Senior Flight Control Engineer
Aerospace Systems Directorate
2130 Eighth Street, WPAFB, OH 45433-7542
COMM 937-938-4617
Email: ba.nguyen@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RQQC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
Approved for Public Release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In the military domain of autonomous aerial refueling operations, automated visual recognition of an approaching aircraft critically supports mission goals. These scholarly articles leverage recent developments in the field of natural image pattern recognition with deep Convolutional Neural Networks (CNNs). The first article reviews the operational details of CNNs, then demonstrates a hyper-parameter optimization process. The second investigates advanced forms of data augmentation in terms of image recognition performance. Finally, the third article demonstrates a novel ensemble confidence measure as well as a modified ensemble compression technique which retains a useful confidence measure in a single student network.

15. SUBJECT TERMS

Deep Learning, Aircraft Identification, Convolutional Neural Network, Autonomy

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Lt Col John Pecarina, AFIT/ENG
U	U	U	U	91	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x3368; john.pecarina@afit.edu