Shape Identification of Space Objects via Light Curve Inversion using Deep Learning Models

Roberto Furfaro
University of Arizona
Richard Linares
Massachusetts Institute of Technology

Vishnu Reddy
University of Arizona

ABSTRACT

Over the past few years, Space Situational Awareness (SSA), generally concerned with acquiring and maintaining knowledge of resident Space Objects (SO) orbiting Earth and potentially the overall cis-lunar space, has become of critical to preventing the loss, disruption, and/or degradation of space capabilities and services. Importantly, threats to operational satellites are also increasing, due to emerging capabilities of potential adversaries. As space becomes more congested and contested, developing a detailed understanding of the SO population became one of the fundamental SSA goals. Currently, the SO catalog includes simplified SO characteristics, e.g. solar radiation pressure and/or drag ballistic coefficients. The currently available simplified description limits the dynamic propagation model used for predicting the catalog to those that assume cannon ball shapes and generic surface properties. Future SO catalogs will have more stringent requirements and shall provide a detailed picture of SO characteristics. An analysis of the current state of the art shows that traditional measurement sources for SO tracking, such as radar and optical, can be employed to extract information on SO characteristics. Such measurements have been shown to be sensitive to SO properties such as shape, attitude, angular velocity, as well as surface parameters.

Recent advancements in deep learning (e.g. Convolutional Neural Networks (CNN), Recurrent Neural networks (RNN)), Generative Adversarial Networks (GAN), Deep Autoencoders (AE)) have demonstrated impressive results in many practical and theoretical fields (e.g. speech recognition, computer vision, robotics). Whereas deep learning methods are becoming ubiquitous in many aspects of our life, they have been barely explored for SSA applications in general and for SO object characterization, in particular. Recently, CNNs have been shown by our research team to be an effective method for SO classification using photometric data. In this paper, we report the results obtained in designing and training a set of deep models capable of retrieving SO shapes from light curves. Traditional shape retrieval methods employ some form of physically-based model inversion. One of the most advanced approach, the Multiple Models Adaptive Estimator (MMAE), runs a bank of Extended Kalman Filters which are based on a set of physical models accounting for different space object properties. The model that minimizes the uncertainty during the retrieval (least residual) is considered to be the model that best represents the SO object properties. Nevertheless, physically-based model inversion is generally ill-posed and computationally expensive. Here, we show how deep learning methods can be employed to provide an effective shape retrieval in a fast and accurate fashion. More specifically, CNN are designed trained and validated for SO shape retrieval. Additionally, we design and train Variational Autoencoders (VAE) to learn the latent distribution of the data. VAE capture the structure of the data and enable an understanding of the complexity of the light curve inversion problem. Indeed, plotting the learned mean and variance and the data distribution enables visual understanding of the ability of the deep network to learn the correct inverse functional relationship between light curves and SO shape.

1. INTRODUCTION

Over the past few years, Space Situational Awareness (SSA) is becoming an area of utmost interest. The latter is due to a dramatic increment of the resident Space Objects (SO) as well as the continuously increasing amount of international

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

space actors, including state and private companies. Consequently, SSA has become an important research topic especially when dealing with the problem of SO identification, characterization and behavior understanding. Enabled by the most recent advancement in sensor technology, researches and operational engineers rely on a large amount of tracking data that can be processed to identify, characterize and understand intention of space objects. The SO catalog maintained by JSpoC currently includes upward of 29,000 SO, with 1,100 of such objects being actively controlled and operated. Researchers working on SSA are interested in providing a detailed understanding of the SO population behavior which must go beyond the currently SO catalog comprising simplified SO characteristics such as solar radiation pressure and drag coefficients. To provide a more realistic and reliable understanding of the SO dynamics, future catalogs that support effective SSA, must include detailed SO characteristics (e.g. shape and state of motion). The latter can be employed in dynamical propagation models to accurately predict SO trajectory and behavior.

Optical sensors are generally employed to track near-geosynchronous SO. Such sensors provide both astrometric and photometric measurements. Consequently, SO properties can be estracted from astrometry pipelines (e.g. trajectories) and photometric data (e.g. shape and state of motion). More specifically, light curves, i.e. flux of photons across a wavelength reflected by the SO and collected by optical sensors, play an important role in determining the SO attitude and state of motion. Indeed, attitude estimation and extraction of other characteristic using light curve data has been demonstrated in Ref. [1, 2, 3, 4, 5].

Traditional measurement sources for SO tracking (e.g. radar and/or optical measurements)have been shown to be sensitive to shape [4, 6], attitude [4], angular velocity [7], and surface parameters [8, 9]. A literature review shows that recent advancement have been made to estimate SO properties. Such techniques heavily rely on estimation theory and include the development of multiple model [4, 10], nonlinear state estimation [11, 12, 7], and full Bayesian inversion [13] approaches for SO characterization. Although grounded in a solid theoretical background, the above mentioned methods tend to be computationally expensive. New techniques are sought that can provide a higher degree of accuracy, computational efficiency and reliability. Recent advancements in machine learning have included deep learning as critical breakthrough technology. Indeed, deep learning methods [14] have shown ground breaking results across a large number of domains. Deep networks are neural networks that comprises more than hidden layers of neurons in their architecture. In such multi-layer neuronal arrangement, deep learning approaches are designed to mimic the function of the brain by learning nonlinear hierarchical features from data that build in abstraction [15]. The latter enabled a higher level of accuracy in typical classification tasks. One of the most popular deep architectures are the so-called Convolutional Neural Networks (CNN). The latter have been used to classify an image or determining the particular content of an image by transforming the original image, through a set of layers with specialized architecture, to a class scores. CNNs have achieved remarkable performance on image processing tasks. Examples include 1) object classification [16], 2) scene classification [17], and 3) video classification [18]. Importantly, the key enabling factor for the success of CNN is the development of techniques that can optimize large scale networks, comprising tens of millions of parameters, as well as massive labeled datasets. More recently, researchers have been exploring the use of deep architecture to find patterns in the data (unsupervised methods). Recently, a class of deep neural networks generally named Auto Encoders (AE) [19] have been devised to find structures on the data. More specifically, deep AE efficiently learn a data representation, usually for dimensionality reduction, by training a deep network in a layerwise fashion to reconstruct the original data. Among the many variations of the AE, a relatively new generation of algorithms called Variational Autoencoders (VAE) [20] have emerged as generative models. Although devised with a AE-like architecture, they have a completely different mathematical formulation: VAEs are directed probabilistic graphical models that learn the posterior distribution of the data using a neural networks. Thus, VAEs are capable of learning the data distribution in the training set and then potentially be employed to generate new data. Applications typically employ data dimensionality reduction, where mean and variance of the distribution are plotted to visualize how data tend to cluster.

In this paper, we explore and evaluate the ability of deep learning algorithms to invert physically-based light curve models and retrieve SOs shape whenever trained on model-generated data. More specifically, we investigate Convolutional Neural Networks (with max-pooling and dropout) [15] as a mean to solve the light curve inverse problem and evaluate the ability to learn the inverse relationship between data and shape. Additionally, pattern data analysis and dimensionality reduction approach are investigated to understand the ability of such algorithms to perform model-based inversion. More specifically, VAEs are trained on light curve data generated by physically-based models to learn the latent distribution over the classes of data. Visualizing the per-class (shape) mean and variance data distribution provides potential tools for understanding how the inverse relationship modeled via CNNs are effective in SO shapes retrieval.

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

Indeed, VAEs enable a cluster inspection/analysis of high-dimensional data. The latter highlights the complexity of the inverse functional relationship between light curves and SO shape (classes). Such complexity is confirmed by non-linear dimensionality reduction techniques, such at the t-Distributed Stochastic Neighbor Embedding (or t-SNE [21]), which projects high-dimensional data in a 2D embedding while preserving the distance between data.

2. INVERSION OF PHYSICALLY-BASED MODELS: DEEP LEARNING MODELS

Shape retrieval from measured light curves is a critical in understanding the nature of SOs. Over the past few years, a modeling approach [22] has emerged where SO light curves can be simulated as function of the object parameters (e.g. shape, size, surface reflectance properties, state of motion). Such *forward models* are specifically devised to describe the functional relationship between SO characteristics and its corresponding light curves. Such models can be classified as physically-based models, because light curves are computed using basic physical principles that integrate state of motion with surface physical properties (e.g. reflectance, absorption). Such models can be potentially inverted, i.e. given the measurements, retrieve the SO properties. More specifically, light curve inversion consists in iteratively adjusting the values of the SO input parameters to the forward model $\boldsymbol{FM}$ until the simulated light curve matches the measured light curves $\boldsymbol{L}$. Formally:

$$\boldsymbol{L} = \boldsymbol{FM}(\boldsymbol{SO}_\theta) + \varepsilon \tag{1}$$

Where $\boldsymbol{SO}_\theta$ represents the SO properties (e.g. shape) and ε accounts for both measurements and model uncertainties, i.e. it represents the adequacy between model and measurements. Generally, inverse problems can be properly solved if they are well-posed. Following Hadamard definition, a problem is well posed if and only if the solution 1) exists, 2) it is unique and 3) depends continuously on the data. However, inverse problems are known to be ill-posed by nature mainly for two reasons [23]. First and foremost, the solution of the inverse problem might not be unique as a set of SO configurations could potentially lead to similar matches between the modeled and measured light curves. Secondly, measurements and model uncertainties may yield large variations in the solution of the inverse problem. Generally, regularization techniques are required to obtain stable as well as reliable solutions of the inverse problems. Here, we consider a data-driven approach where the inversion of physically-based light curve models is obtained by designing deep networks that can learn the inverse relationship between simulated light curves and SO shape. We tackle a challenging problem where we train a CNN to discriminate between rocket bodies with similar shape. Importantly, we show that VAE can be employed to learn the distribution of the simulated datasets, which provides an evaluation of the nature of the inversion.

2.1 Convolutional Neural Networks

Over the past few years, there has been an explosion of machine learning algorithms. Such algorithms can learn from data to accomplish specific tasks (e.g. image recognition, object identification, natural language process, etc.). Among the various available techniques, deep learning, comprising methods and techniques to design and train multi-layer neural networks, has been playing a dominant role. In contrast to shallow networks, deep networks refers to a class of neural networks comprising a number of hidden layers greater than one. Among all possible systems, one of the most powerful deep architectures is called Convolutional Neural Networks (CNN). Generally, the architecture of a CNN is designed to take advantage of the 2D or 3D $[width, height, depth]$ structure of an input image which is processed as pixels values. The basic CNN structure uses many types of layers which are 1) Convolutional Layer, 2) Pooling Layer, 3) Fully Connected Layer and 4) Output Layer. The core layer, i.e. the Convolutional layer, extract features from the input volume by applying filters on the image. It is the most demanding layer in terms of computations of a CNN and the layer′s parameters consist of a set of learnable filters. Each filter is basically a matrix spatially smaller than the image which is scanned along width and height (2D case). Importantly, filters (or kernels) are the weights of this layer. In fact, as the filter is sliding on the input image, it multiplies its values with the original pixel values of the image and these multiplications are all summed up giving only one number as output. Repeating this procedure for all the regions on which filter is applied, the input volume is reduced and transformed and then passed to the *Max-pooling layers*. Matematically, the convolutional layer can be described as follows:

$$(X * Y)(i, j) = \sum_{n=0}^{N} \sum_{m=0}^{M} X_{m,n} * W_{i-m, j-n} \tag{2}$$

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

Here, W is the convolution kernel corresponding to the randomly initialized weights, and X is the image with indices (m,n). CNN typically employs the nonlinear activation function called *ReLU* function (i.e., Rectified Linear Unit) described as $f(x) = max(0,x)$. Spatial pooling layers group local features from spatially adjacent pixels to improve robustness. A set of convolutional layers are generally stacked below a fully connected layer that feeds a soft-max layer ([24]), which outputs the probability of the image belonging to one of the classes. CNN are easier to train due to inherent parameter sharing in the convolutional layer. For a classification task, the cross-entropy function [24] is generally employed as cost to minimize.CNNs are trained in batch mode via Stochastic Gradient Descent (SDG) with variable size of mini-batches. The dropout technique improves generalization and avoids overfitting.

2.2 Variational Autoencoders for Unsupervised Learning

Variational AutoEncoders (VAE) [20] are a more recent version of the classical autoencoders (AE). The latter are neural networks constructed by stacking layers of neurons that can perform data reconstruction using an encoder-decoder approach. VAE performs both encoding and decoding functions by imposing additional constraints on the encoded representation. At the very basic level, such constraints make the VAE an algorithm capable of learning the latent variable model behind the input data. Traditional AE generally learn an arbitrary function to represent the encode-decode operation. Conversely, VAE learn the parameters that describe the probability distribution that models the data (encoding function). Once such distribution is learned from the data, one can sample the parameters space such that the encoding portion of the VAE can generate samples closely resembling the training sets (i.e. works as a generative model). VAEs generally rely on the assumption that one can potentially sample the input data from a Gaussian distribution of latent parameters, or in other words, it assumes that the data are normally distributed. The generative model is assumed to be a deep latent Gaussian model. More specifically, let x be a local (observed) variable and let z be the corresponding latent variable, with joint distribution described as follows:

$$p_\theta(x,z) = p_\theta(x|z)p(z) \tag{3}$$

In typical setting for Bayesian modelling, one assumes that the VAE has a single layer of latent variables with normal prior distribution in z. As an example, one can represent the decoder network as multivariate Bernoulli distribution with a network represented as follows:

$$p_\theta(x|z) = Bern(\sigma(W_2 * h(W_1 z + b_1) + b_2)) \tag{4}$$

Where σ and h are the sigmoid fucntion and some non-linear function. respectively and $\theta = W_1, W_2, b_1, b_2$ are the networks learnable parameters. Once the generative process is specified, latent variables z and model parameters θ must be inferred. Due to analytical intractability of the inference process, a typical optimization process known as Variational Inference is generally employed. The process seeks to approximate the posterior of the latent variable z given the data, i.e. $q_\theta(z|x)$ that minimize the Kullback-Leibler (KL) divergence between true and approximate posteriors:

$$\phi^* = argmin_\phi KL[q_\phi(z|x)||p_\theta(z|x)] \tag{5}$$

Although a sound optimization problem, due to numerical stability and better tractability, one generally prefers to maximize an alternative objective function named Evidence Lower Bound (ELBO), expressed as:

$$ELBO(q) = \mathbb{E}_{q_\phi(z|x)}[log(p_\theta(x|z)] - KL[q_\phi(z|x)||p(z)] \tag{6}$$

Importantly, $q_\phi(z|x)$ can be seen as a probabilistic encoder, i.e. given an observation x, it encodes it into a distribution over an hidden lower-dimensional representation. At this stage, one can perform training via gradient-base optimization of ELBO with respect to the model parameters θ and the variational parameters ϕ. The latter require knwoledge of the gradient which is generally hard to find analytically. Monte Carlo-based estimation methods are the dominant methods for implementation in actual codes. The idea is to write the gradient of the objective function ELBO as an expectation of the gradient and approximate it via Monte Carlo sampling. Once the gradient estimate is available one can perform standard stochastic gradient descent.

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

2.3 Statistical Algorithms for Clustering: t-SNE

The t-Distributed Stochastic Neighbor Embedding [21] is a probabilistic technique particularly suitable for visualization of high-dimensional data. The overall algorithm is designed to minimize the divergence between two distributions, i.e. between a) a distribution that measures the similarities between two input points pairwise, and b) a distribution that measures similarities of the corresponding points in the embedding as mapped pairwise on the low-dimensional subspace. The overall idea is to embed points living in a high-dimensional space into a low-dimensional space (generally 2-D or 3-D) in such a way that the mapping preserves similarities (distance) between points. Consequently, points that are nearby in the high-dimensional space are expected to be close in the low-dimensional embedding. As such, one can visualize points in low-dimensional spaces to find natural clustering that occur in the high-dimensional spaces. The input data to the algorithm is the matrix $X = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}\} \in \mathbb{R}^{\mathbf{D}\times\mathbf{N}}$ which contains N vectors $x_i \in \mathbb{R}^D$. After some preprocessing (e.g. subtraction of the mean and division by the standard deviation) the algorithm computes the distance $d(x_j, x_k)$ between each pair of points comprising the matrix X. The distance is then used to compute the *Perplexity*. The latter is generally defined in terms of a model Gaussian distribution. Indeed, one defines the conditional probability of k given j as follows:

$$p(k|i) = \frac{\exp(-d(x_j,x_k)^2/(2\sigma_j^2))}{\sum_{k\neq j}\exp(-d(x_j,x_k)^2/(2\sigma_j^2)} \tag{7}$$

and

$$p(j|j) = 0 \tag{8}$$

Define now the joint probability p_{jk} as follows:

$$p_{ij} = \frac{p(j|k)+p(k|j)}{2N} \tag{9}$$

and the Shannon Entropy of P_j as follows:

$$H(P_j) = -\sum_k p(k|j)\log_2(p(k|j)) \tag{10}$$

Where the term P_j represents the conditional probability distribution over all data points x_i. The *Perplexity* $Perp(P_j)$, which measures the number of neighbors of point j, is defined as follows:

$$Perp(P_j) = 2^{H(P_j)} \tag{11}$$

Generally, the embedding of points contained in the matrix X into a low-dimensional space, the tSNE algorithm executes an opimization procedure,i.e. it attempts to minimize the *Kullback-Leibler* (KL) divergence between the model Gaussian distribution of the points in X and a student t distribution of point Z in the selected low-dimensional space. Given the points $Z = \{z_1, z_2, ..., z_N\}$ in the selected low-dimensional embedding, the probability model of the distribution of the distances between two points can be described as follows:

$$q_{jk} = \frac{(1+\|z_j-z_k\|^2)^{-}1}{\sum_m\sum_{n\neq m}(1+\|z_m-z_n\|^2)^{-}1} \tag{12}$$

The KL divergence between th two distributions is therefore:

$$KL(P\|Q) = \sum_{j\neq k} p_{jk}\log\frac{p_jk}{q_{jk}} \tag{13}$$

The minimization of the KL divergent is commonly executed by implementing a gradient descent approach.

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

3. MODEL INVERSION VIA DEEP LEARNING: MODEL DESCRIPTION AND TRAINING SET GENERATION

Constructing a deep network capable of learning the inverse relationship between lightcurves and SO shapes, requires a physically-based forward model. The model describes an arbitrary SO shape as a finite number of flat triangular plates each equipped with specified reflectance properties. The model is also equipped with dynamical equations of the SO which describe the trajectory and attitude state while progressing along a specified orbit. The training set is generated by running the forward model for different shape sonfigurations and variable stae of motion.

3.1 Physically-based Models:Shape Model Definition

The shape model considered in this paper consists of a finite number of flat facets, where each facet has a set of basis vectors associated with it. These basis vectors are defined in Figure 1 and consist of three unit vectors $\mathbf{u}_n^B$, $\mathbf{u}_u^B$, and $\mathbf{u}_v^B$. The unit vector $\mathbf{u}_n^B$ points in the direction of the outward normal to the facet. For convex surfaces, this model becomes more accurate as the number of facets is increased. The vectors $\mathbf{u}_u^B$ and $\mathbf{u}_v^B$ are in the plane of the facet. The SO are assumed to be rigid bodies and therefore, the unit vectors $\mathbf{u}_n^B$, $\mathbf{u}_u^B$ and $\mathbf{u}_v^B$ do not change since they are expressed in the body frame.

The light curve and the SRP models discussed in the next sections require that these vectors be expressed in inertial coordinates and since the SO body is rotating, these vectors will change with respect to the inertial frame. The body vectors can be rotated to the inertial frame by the standard attitude mapping given by:

$$\mathbf{u}_k^B = A(\mathbf{q}_I^B)\mathbf{u}_k^I, \quad k = u, v, n \tag{14}$$

where $A(\mathbf{q}_I^B)$ is the attitude matrix mapping the inertial frame to the body frame using the quaternion parameterization. Furthermore, the unit vector $\mathbf{u}_{\text{sun}}^I$ points from the SO to the Sun direction and the unit vector $\mathbf{u}_{\text{obs}}^I$ points from the SO to the observer. The vector $\mathbf{u}_h^I$ is the normalized half vector between $\mathbf{u}_{\text{sun}}^I$ and $\mathbf{u}_{\text{obs}}^I$. This vector is also known as the Sun-SO-Observer bisector. Each facet has an area $\mathscr{A}(i)$ associated with it. Once the number of facets has been defined and their basis vectors are known, the areas $\mathscr{A}(i)$ define the size and shape of the SO. To determine the SRP forces and light curve characteristics, the surface properties must be defined for each facet. The shape model used for this work use triangular facets defined by the location of their vertices $\mathbf{b}_i$. Then the area of the i^{th} triangular facet formed by the convex hull of the control points is given by, $\mathscr{A}(i) = \|\mathbf{d}(i) \times \mathbf{l}(i)\|$, where $\mathbf{d}(i)$ and $\mathbf{l}(i)$ are the vectors defining two sides of the facets or $\mathbf{d}(i) = \mathbf{b}_i - \mathbf{b}_{i-1}$, $\mathbf{l}(i) = \mathbf{b}_i - \mathbf{b}_{i+1}$. The unit normal vector is given by

$$\mathbf{u}_n = \frac{\mathbf{d}(i) \times \mathbf{l}(i)}{\|\mathbf{d}(i) \times \mathbf{l}(i)\|} \tag{15}$$

For this work it is assumed that each facet has the same material parameters (specular coefficients, diffuse coefficients, and other reflection parameters discussed in the next section).

3.2 Physically-based Models: Lightcurves Generation

There are a number of models used for simulating light curve measurements in literature and Ref. wetterer2013 provides a good summary of the most popular ones adopted for SO applications. These models differ in the physics that they represent and their level of complexity, but for SO applications the ability to model specular reflection and complex shapes while converting energy is desirable. The Ashikhmin-Shirley[25] (AS) model has all the desirable properties while producing realistic SO light curve. This model is based on the bidirectional reflectance distribution function (BRDF) which models light distribution scattered from the surface due to the incident light. The BRDF at any point on the surface is a function of two directions, the direction from which the light source originates and the direction from which the scattered light leaves the observed surface. The model in Ref. Ashikhmin:00 decomposes the BRDF into a specular component and a diffuse component. The two terms sum to give the total BRDF:

$$f_r = (dR_d + sR_s) \tag{16}$$

which depends on the diffuse bidirectional reflectance (R_d) and the specular bidirectional reflectance (R_s) and the fraction of each to the total (d and s respectively where $d + s = 1$). Where i denotes the i^{th} facet of the SO. Each facet contributes independently to the brightness and total brightness is the sum over each facet's contribution. The diffuse component represents light that is scattered equally in all directions (Lambertian) and the specular component

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

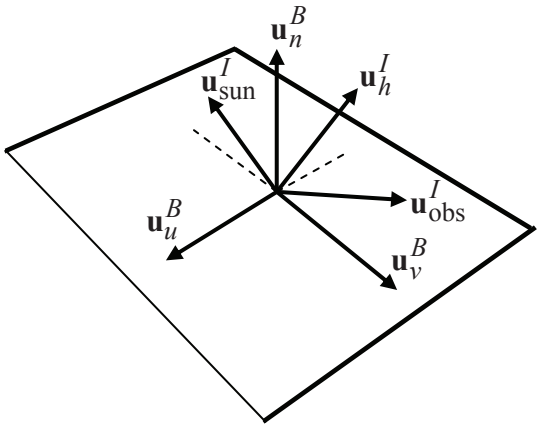


Fig. 1: Reflection Geometry

represents light that is concentrated about some direction (mirror-like). Reference Ashikhmin:00 develops a model for continuous arbitrary surfaces but simplifies for flat surfaces. This simplified model is employed in this work as shape models are considered to consist of a finite number of flat facets. Therefore the total observed brightness of an object becomes the sum of the contribution from each facet. In each model, however, $c = \mathbf{V}^T\mathbf{H}$, ρ is the diffuse reflectance ($0 \leq \rho \leq 1$), and F_0 is the specular reflectance of the surface at normal incidence ($0 \leq F_0 \leq 1$). To be used as a prediction tool for brightness and radiation pressure calculations, an important aspect of the BRDF is energy conservation. For energy to be conserved, the integral of the BRDF times $\cos(\theta_r)$ over all solid angles in the hemisphere with $\theta_r \leq 90$ needs to be less than unity, with

$$\int_0^{2\pi}\int_0^{\pi/2} f_r \cos(\theta_r)\sin(\theta_r)\,d\theta_r d\phi = R_d + R_s \tag{17}$$

For the BRDF given in Eq. (16), this corresponds to constant values of $R_d = \rho d$ and $R_s = sF_0$. The remaining energy not reflected by the surface is either transmitted or absorbed. In this paper it is assumed the transmitted energy is zero. The diffuse bidirectional reflectance is then calculated as follows:

$$R_d = \frac{28\rho}{23\pi}(1-sF_0)\left(1-\left(1-\frac{\mathbf{N}^T\mathbf{L}}{2}\right)^5\right)\cdot \left(1-\left(1-\frac{\mathbf{N}^T\mathbf{V}}{2}\right)^5\right) \tag{18}$$

where

$$F = F_0 + \left(\frac{1}{s} - F_0\right)(1-c)^5 \tag{19}$$

The vectors $\mathbf{L}(t_i)$ and $\mathbf{V}(t_i)$ denote the unit vector from the SO to the Sun and the unit vector from the SO to the observer, respectively, and together they define the observation geometry (shown in Figure 1). In addition to d, ρ, and F_0, the Ashikhmin-Shirley BRDF has two exponential factors (n_u, n_v) that define the reflectance properties of each surface. The Ashikhmin-Shirley diffuse and specular reflectivities are not constant but rather complicated functions of illumination angle, exponential factor, and the diffuse and specular reflectances. In all cases, however, $R_d + R_s \leq 1$, so energy is conserved. The parameters of the Phong model that dictate the directional (local horizontal or vertical) distribution of the specular terms are n_u and n_v. The specular bidirectional reflectance for the AS model is given by

$$R_s = \frac{F\sqrt{(n_u+1)(n_v+1)}}{8c\pi\max[\mathbf{N}^T\mathbf{L},\mathbf{NV}]}(\cos(\alpha))^\gamma \tag{20}$$

where $\gamma = n_u\cos^2(\beta) + n_v\sin^2(\beta)$.

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

The apparent magnitude of an SO is the result of sunlight reflecting off of its surfaces along the line-of-sight to an observer. First, the fraction of visible sunlight that strikes an object (and is not absorbed) is computed by

$$F_{\text{sun}}(i) = C_{\text{sun,vis}} \left(\mathbf{u}_n^I(i) \cdot \mathbf{u}_{\text{sun}}^I\right) \tag{21}$$

where $C_{\text{sun,vis}} = 1062 \text{ W/m}^2$ is the power per square meter impinging on a given object due to visible light striking the surface. If either the angle between the surface normal and the observer's direction or the angle between the surface normal and Sun direction is greater than $\pi/2$ then there is no light reflected toward the observer. If this is the case then the fraction of visible light is set to $F_{\text{sun}}(i) = 0$. Next, the fraction of sunlight that strikes an object that is reflected must be computed:

$$F_{\text{obs}}(i) = \frac{F_{\text{sun}}(i)\rho_{\text{total}}(i)\mathscr{A}(i)\left(\mathbf{u}_n^I(i) \cdot \mathbf{u}_{\text{obs}}^I\right)}{\|\mathbf{d}^I\|^2} \tag{22}$$

The reflected light of each facet is now used to compute the total photon flux, which is measured by an observer:

$$\tilde{F} = \left[\sum_{i=1}^{N} F_{\text{obs}}(i)\right] + v_{\text{CDD}} \tag{23}$$

where v_{CDD} is the measurement noise associated with flux measured by a Charge Coupled Device (CCD) sensor. The total photon flux is then used to compute the apparent brightness magnitude

$$m_{\text{app}} = -26.7 - 2.5\log_{10}\left|\frac{\tilde{F}}{C_{\text{sun,vis}}}\right| \tag{24}$$

where -26.7 is the apparent magnitude of the Sun.

A number of parameterizations exist to specify attitude, including Euler angles, quaternions, and Rodrigues parameters.[26] This paper uses the quaternion, which is based on the Euler angle/axis parameterization. The quaternion is defined as $\mathbf{q} \equiv [\boldsymbol{\rho}^T \quad q_4]^T$ with $\boldsymbol{\rho} = \hat{\mathbf{e}}\sin(\nu/2)$, and $q_4 = \cos(\nu/2)$, where $\hat{\mathbf{e}}$ and ν are the Euler axis of rotation and rotation angle, respectively. Clearly, the quaternion must satisfy a unit norm constraint, $\mathbf{q}^T\mathbf{q} = 1$. In terms of the quaternion, the attitude matrix is given by

$$A(\mathbf{q}) = \Xi^T(\mathbf{q})\Psi(\mathbf{q}) \tag{25}$$

where

$$\Xi(\mathbf{q}) \equiv \begin{bmatrix} q_4 I_{3\times 3} + [\boldsymbol{\rho}\times] \\ -\boldsymbol{\rho}^T \end{bmatrix} \tag{26a}$$

$$\Psi(\mathbf{q}) \equiv \begin{bmatrix} q_4 I_{3\times 3} - [\boldsymbol{\rho}\times] \\ -\boldsymbol{\rho}^T \end{bmatrix} \tag{26b}$$

with

$$[\mathbf{g}\times] \equiv \begin{bmatrix} 0 & -g_3 & g_2 \\ g_3 & 0 & -g_1 \\ -g_2 & g_1 & 0 \end{bmatrix} \tag{27}$$

for any general 3×1 vector $\mathbf{g}$ defined such that $[\mathbf{g}\times]\mathbf{b} = \mathbf{g} \times \mathbf{b}$.

The rotational dynamics are given by the coupled first order differential equations:

$$\dot{\mathbf{q}}_I^B = \frac{1}{2}\Xi(\mathbf{q}_I^B)\boldsymbol{\omega}_{B/I}^B \tag{28a}$$

$$\dot{\boldsymbol{\omega}}_{B/I}^B = J_{\text{SO}}^{-1}\left(\mathbf{T} + \mathbf{T}_{\text{srp}}^B - \left[\boldsymbol{\omega}_{B/I}^B\times\right] J_{\text{SO}}\boldsymbol{\omega}_{B/I}^B\right) \tag{28b}$$

where $\boldsymbol{\omega}_{B/I}^B$ is the angular velocity of the SO with respect to the inertial frame, expressed in body coordinates, J_{SO} is the inertia matrix of the SO. The vectors $\mathbf{T}_{\text{srp}}^B$ and $\mathbf{T}$ are the net torques acting on the SO due to SRP expressed in body coordinates and the control torque, respectively.

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

3.3 Training Set generation: Simulating Labeled Light Curves

For the simulated case, the labeled training data samples are generated using the light curve model discussed above. The parameters required to define the AS light curve model are sampled. Here, we considered four shapes all belonging to the rocket body classes with different characteristics. As shown in Figure 2,the four classes are modelled as 1) A cylinder with round top, 2) a simple cylinder, 3) a shape representative of an Atlas Upper Stage; and 4) a shape representative of a Falcon 9 Upper Stage. Note that there is a substantial difference is nozzle size between the Atlas and the Falcon 9 upper stages. Importantly, rocket body models are generated using octant triangulation of a sphere discussed in Ref. [27] which divided the surface of a sphere into N facet normal. Then rocket body models are generated by connecting two hemisphere ends of radius r with cylinder of height l. This model is not exact for all rocket bodies but is close enough to approximate the types of light curves seen for rocket bodies.

$$\begin{aligned} J_{\text{rocket}} = m_{\text{SO}} \Bigg\{ & \frac{V_{\text{cyl}}}{V_{\text{tot}}} \text{diag}\left[\frac{1}{12}(3r^2+l^2), \frac{1}{12}(3r^2+l^2), \frac{r^2}{2}\right] \\ & + \frac{V_{\text{top}}}{V_{\text{tot}}} \text{diag}\left[\frac{1}{12}(3r^2+l^2), \frac{1}{12}(3r^2+l^2), \frac{r^2}{2}\right] \\ & + \left(\frac{V_{\text{top}}}{V_{\text{tot}}}\left(\frac{l}{2}+\frac{3r}{8}\right) + \frac{V_{\text{cyl}}}{V_{\text{tot}}}\left(\frac{l}{2}-\frac{3r}{8}\right)\right)\left(I_{3\times 3} - \mathbf{e}\mathbf{e}^T\right) \\ & + 2\frac{V_{\text{top}}}{V_{\text{tot}}} r^2 \text{diag}\left[\frac{83}{320}, \frac{83}{320}, \frac{2}{5}\right] \Bigg\} \end{aligned} \tag{29}$$

Where $\mathbf{e} = [0,0,1]^T$ and the volume of the top hemisphere is given by $V_{\text{top}} = 2/3\pi r^3$ and it is assumed the bottom volume is $V_{\text{bot}} = V_{\text{top}}$. The volume of the cylinder is given by $V_{\text{cyl}} = \pi r^2 l$ and the total volume is $V_{\text{tot}} = V_{\text{top}} + V_{\text{bot}} + V_{\text{cyl}}$.

The model has been employed to generate a set of light curves associated to each of the four considered classes. The simulation conditions are reported as follows:

- Geographic position of the ground site is 0° North, 172° West with 0 km altitude
- The orbital elements are given by $a = 25864.16932$ km, $e = 0.743$, $i = 30.0083$ deg, $\omega = \Omega = 0.0$ deg and $M_0 = 91.065$ deg
- The initial time of the simulation is May 8, 2020 at 5:27.55 UTC

For all simulation scenarios,initial attitude has been represented by a set of quaternions sampled out of an uniform distribution. Additionally, the components of the angular velocity vector have been sampled out of a uniform distribution with magnitude $0.0014 rad/sec$. Measurements of apparent magnitude are produced using zero-mean white-noise error processes with a standard deviation of 0.05 for magnitude. The initial errors for the states are 50 deg for all three attitudes, $1,000$ deg/hr for the rotational rate. The time interval between the measurements is set to 5 seconds and data are simulated for 1 hour. Figure 3 shows the simulated magnitude measurements for the four rocket body classes.

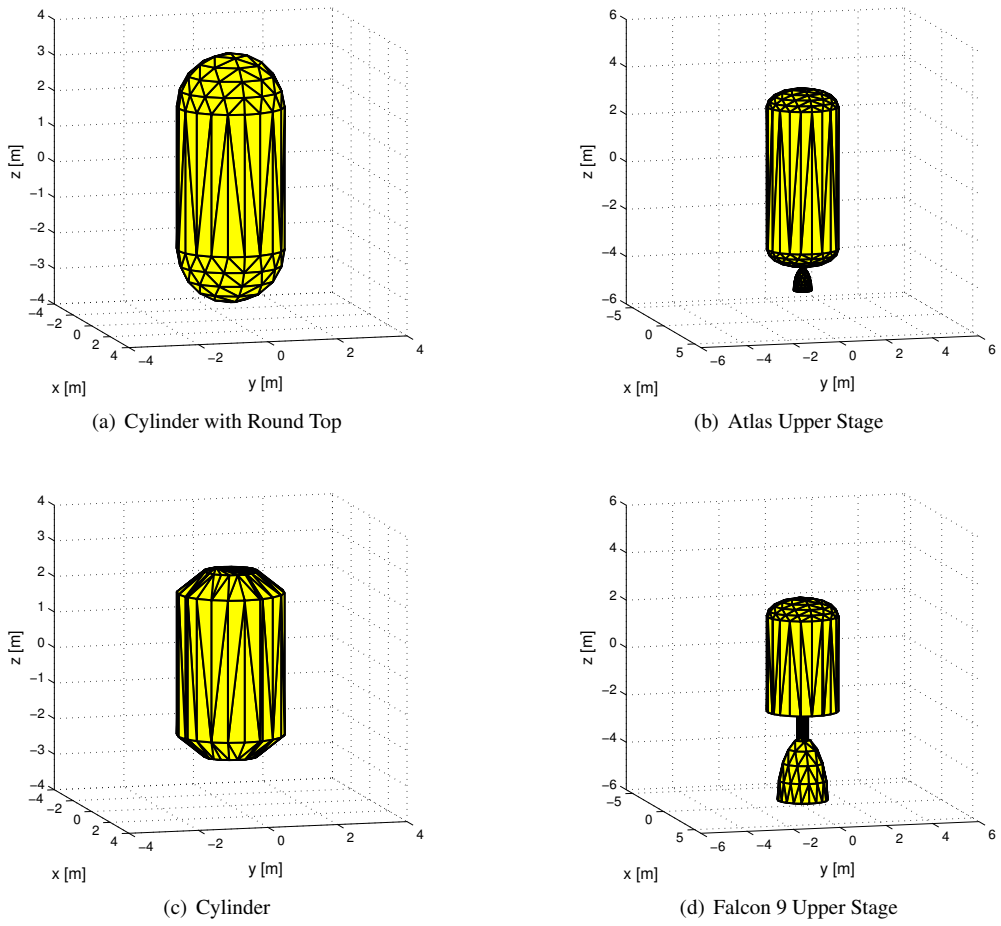
4. RESULTS

4.1 CNN Design, Training and Classification for Shape Determination

As previously discussed, the training data set consists of pairs of simulated light curve measurement vectors and class vectors. Here, The input light curve and output class vector are denoted by $\mathbf{x} \in \mathbb{R}^{1\times m}$ and $\mathbf{y} \in \mathbb{R}^{1\times n_c}$, respectively, where m and n_c denote the number of light curve measurements and number of classes, respectively. Here,a CNN is trained to map the measurement vector, $\mathbf{x}$, to classes, $\mathbf{y}$, using a set of training examples. The CNN designed to learn the functional relationship between simulated measurements and SO classes (shapes) consists of 1-D convolutional layers with rectified linear unit (ReLU) activation, dropout, max-pooling, and two fully connected layer with ReLU activation (Figure 4). The output layer uses softmax function to map to classification states. Each convolutional layer has the following form:

$$\mathbf{h}^{cov}(\mathbf{x}) = \mathbf{f}(\mathbf{x} * W^{cov} + \mathbf{b}^{cov}) \tag{30}$$

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com



(a) Cylinder with Round Top

(b) Atlas Upper Stage

(c) Cylinder

(d) Falcon 9 Upper Stage

Fig. 2: Representative Shape Models for Rocket Bodies

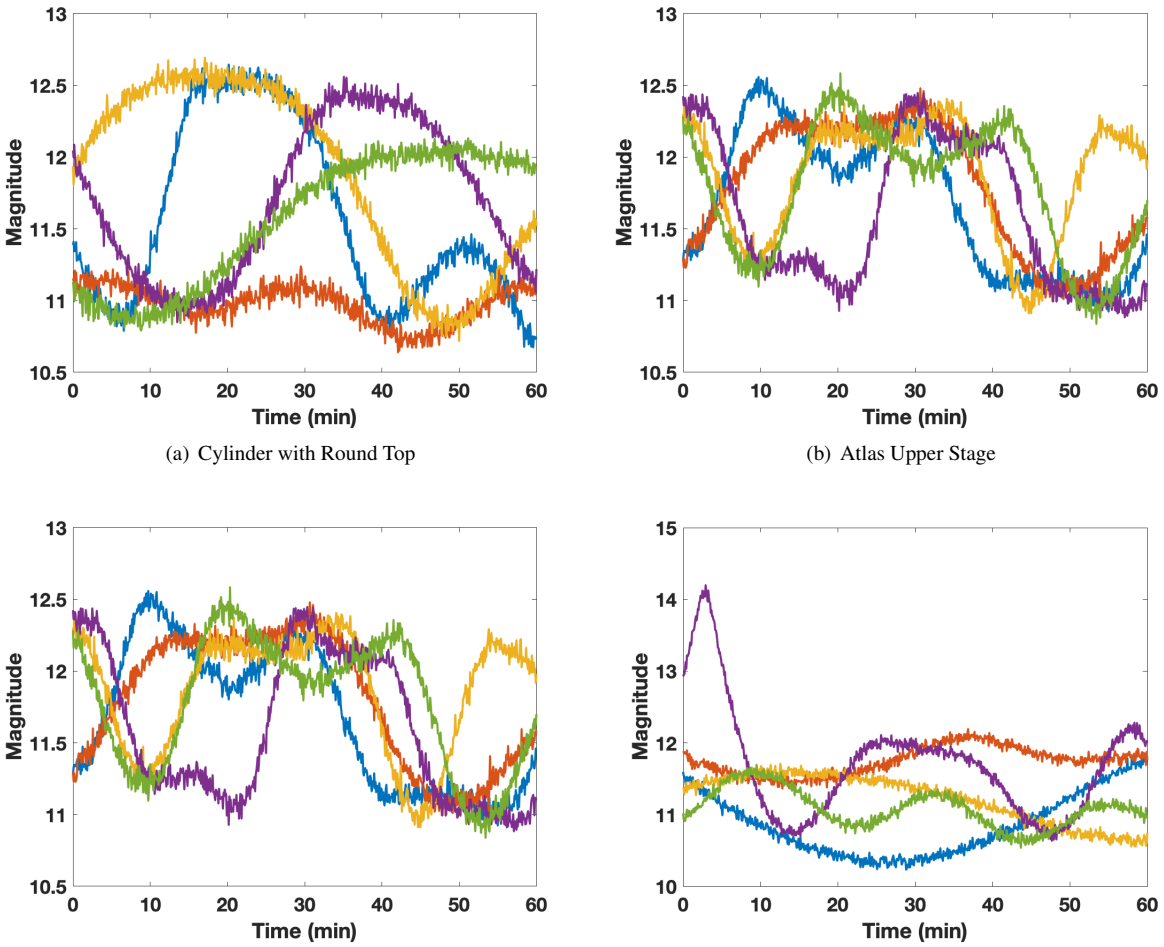
Where $*$ denotes the convolution operator, W^{cov} denotes the convolution kernel, and $\mathbf{f}$ is the activation function for each layer that adds nonlinearity to the feature vector. This work uses ReLU for convolutional layer activation. The ReLU function is given by $\mathbf{f}(\mathbf{x}) = \max(\mathbf{0}, \mathbf{x})$, where it is zero for negative input values and linear for positive input values. The convolution of $\mathbf{x}$ with W^{cov} defines the output feature map $\mathbf{h}^{cov}(\mathbf{x})$. The number of output maps is determined by the number of convolution filters for each convolutional layer. Each convolution layer has a collection of kernels of given size that are learned directly from the data. For the light curve problem the convolutions are of one dimensional time-series data. Then for input vectors having size $(1, s_x)$ the output vector is given by $(1, s_y)$ and the output vector size can then be calculated from the size of the kernel, s_k, and is given by $s_y = s_x - s_k + 1$. After the convolution is applied the output vector is reduced in size but zero padding is used in this work to keep the size of the feature vectors constant through each convolutional layers.

Then a CNN applies a series of these kernels W^{cov} in a layered fashion where each layer has a different size kernel that learns features on a given scale. To simplify the information in the output of each convolutional layer, max-pooling is used. In this work max-pooling with 1×4 kernel between each convolution layer is used. The max-pooling operation convolves the 1×4 kernel over the output of each convolutional layer returning the max of the outputs over the 1×4 region. Finally, at the final layer a nonlinear function is applied to the output using a traditional neural network. This final layer uses a fully connected neural network layer and is given by

$$\mathbf{h}^{fc}(\mathbf{x}) = \mathbf{f}\left(W^{fc}\mathbf{x} + \mathbf{b}^{fc}\right) \tag{31}$$

After the fully connected layer a softmax function is used to provide outputs in the ranges $(0, 1)$ that add up to 1. The

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com



(a) Cylinder with Round Top

(b) Atlas Upper Stage

(c) Cylinder

(d) Falcon 9 Upper Stage

Fig. 3: Apparent magnitude for samples of the four classes of rocket bodies

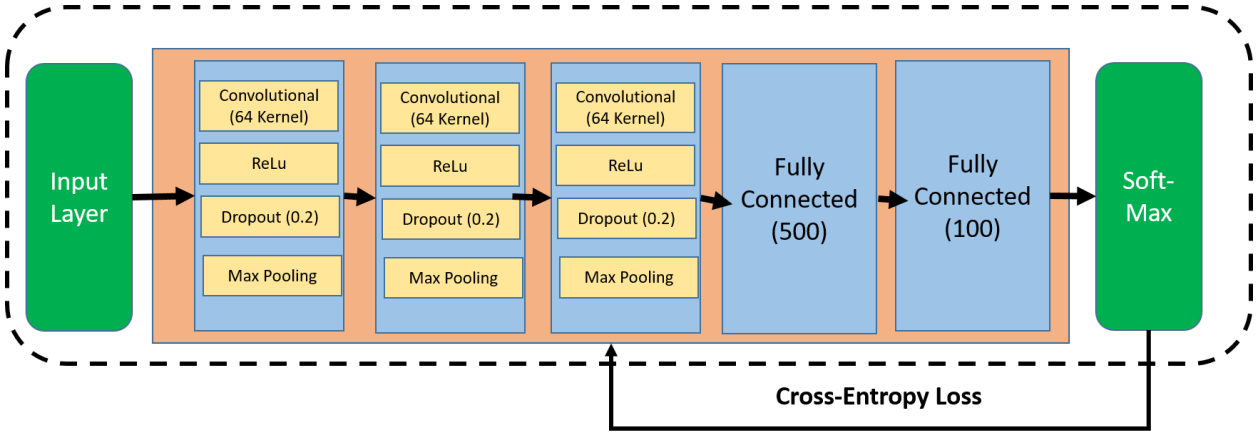


Fig. 4: Network Architecture

softmax function is defined by

$$y_j\left(\mathbf{h}^{fc}(\mathbf{x})\right) = \frac{\exp(h_j^{fc})}{\sum_i \exp(h_i^{fc})} \tag{32}$$

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

The convolutional kernel and fully connected output layer parameters are cased into the vector $\boldsymbol{\theta}$. The cost function used for this work is the cross-entropy loss. This loss function minimizes the cross-entropy loss between training outputs and the CNN outputs, and is given by:

$$\begin{aligned} L(\boldsymbol{\theta}) &= \frac{1}{N}\sum_{i=1}^{N} H(\mathbf{y},\tilde{\mathbf{y}}) \\ &= -\frac{1}{N}\sum_{i=1}^{N} [\mathbf{y}\log\tilde{\mathbf{y}} + (1-\mathbf{y})\log(1-\tilde{\mathbf{y}})] \end{aligned} \tag{33}$$

where $\tilde{\mathbf{y}}$ are the training examples and $\mathbf{y}$ are the outputs from the CNN. Then the CNN classification approach is trained by stochastic gradient descent by minimizing the cross-entropy loss from the outputs compared to the labelled data. Figure 4 shows the full architecture of the network used for this work. LeCun [24] showed that stochastic online learning is superior against the full batch mode as it is faster and results in more accurate solutions. The weights for the output layer and the convolutional layer are updated using the following relationship

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) + \eta\frac{\partial L}{\partial \boldsymbol{\theta}} \tag{34}$$

where t denotes the iteration step and η is the learning rate. The $\frac{\partial L}{\partial \boldsymbol{\theta}}$ is the gradient of the loss with respect to the overall network parameters. This update is calculated for small batches over the entire training sets. Using the small batches allows for small updates to the gradient while reducing the noise in the gradient of individual training samples. This method of updating the parameters is referred to as stochastic gradient descent and the gradient is calculated with error back propagation.

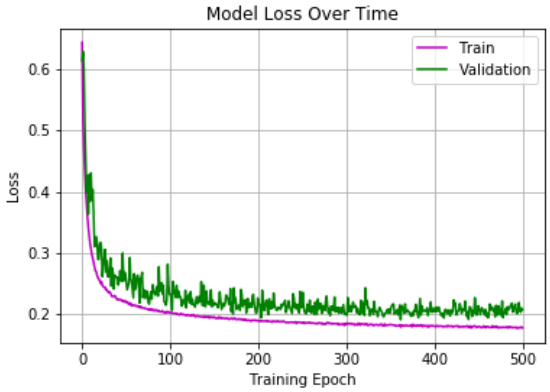
The training set is generated according to the simulation conditions reported in section 3.3 and comprises $200,000$ light curves associated with four (4) possible classes for the four rocket body shapes (i.e. cylinder with round top, simple cylinder, Atlas and Falcon 9 upper stages). Here, we split the training set according to the usual 80/20 rule and use $160,000$ light curves to train the CNN. The remaining $40,000$ curves are employed to validate the CNN during training to avoid overfitting.

The Keras (Python) library wit Tensorflow[28] as backend are employed to design and train the network. The proposed CNN architecture is shown in 4. The CNN comprises a seven (7) layer structure, including input layer, three convolutional layers, two fully connected layers and one softmax layer. The input layers manages a light curve input vector comprising 182 data points. The convolutional layers have 64, 32 and 64 filters, respectively. Both max-pooling (1×2 pooling kernel) and dropout are applied after each convolutional layer.Dropout regularization rates are set to be 0.2 for the convolutional layers and 0.5 for fully connected layers. Training occurred for 500 epochs using stochastic gradient descent and mini-batch of 256 samples. Importantly, the 1D-CNN has been trained using a single GPU with 1500 processors. Figure 6 shows the results of the training process and the overall performances. Figure 5(a) shows the behavior of the cross-entropy loss as function of the epoch. Figure 5(b) shows the model accuracy as function of the epoch. Both test and training sets have a similar accuracy result. We report that the CNN exhibits an accuracy of 92% on the test set. Training time is reported to be $11,000$ sec. Figure 7 reports the confusion matrix associated with the test set comprising $40,000$ samples on which the CNN is not trained and where the CNN predictions are compared with the targeted samples. Importantly, the highest error occurs on mislabeling between class 1 (Cylinder with round top) and class 4 (Falcon 9 upper stage). Here, the CNN interprets the Falcon 9 upper stage as cylinder 797 case where the target is class 4, with about 2% contribution to the global CNN error.

4.1.1 VAE for Shape Determination: Unsupervised Learning

Understanding the training data structure may provide insight into the ability of the deep network to effectively learn the relationship between light curves and shape thus providing an effective solution to the inverse problem. Therefore, understanding the distribution of the data generated via a physically-based forward model may help evaluate the effectiveness of the CNN. Pattern in data can be potentially discovered with visualization aids. However, the visualization of high dimensional data sets is generally a difficult problem. Indeed, the measured light curves employed in the training set are represented as one-dimensional vectors comprising 721 components. A conventional approach to data reduction and visualization considers processing the data using PCA, where the most significant two or three

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com



(a) CNN Loss

(b) CNN Accuracy

Fig. 5: CNN Classifications Results

	Precision	Recall	F-1
Cylinder with RT	0.88	0.95	0.92
Cylinder	0.92	0.98	0.95
Atlas Upper Stage	0.95	0.88	0.91
Falcon 9 Upper Stage	0.94	0.88	0.91
Accuracy			0.92
Macro avg	0.92	0.92	0.92
Weighted avg	0.92	0.92	0.92

(a) CNN Overall Performances

Fig. 6: CNN Classifications Results

components can be visualized in two-dimensional and three dimensional spaces. However, as a linear technique, PCA captures only the linear structures in the data and generally linearly maps high dimensional spaces into low dimensional subspaces. To avoid this pitfall, especially for unknown, arbitrarily complex representations of non-linear relationships, we train VAEs on the data to learn their intrisic latent distribution. VAEs are shown to capture the latent distribution via the non-linear representation power of neural networks. The VAE architecture employed in this study is reported in figure 8. Here, we considered a single-layer VAE for both decoder and encoder. As explained earlier, the decoder represents the generative model where the data distribution is represented by the probability distribution

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

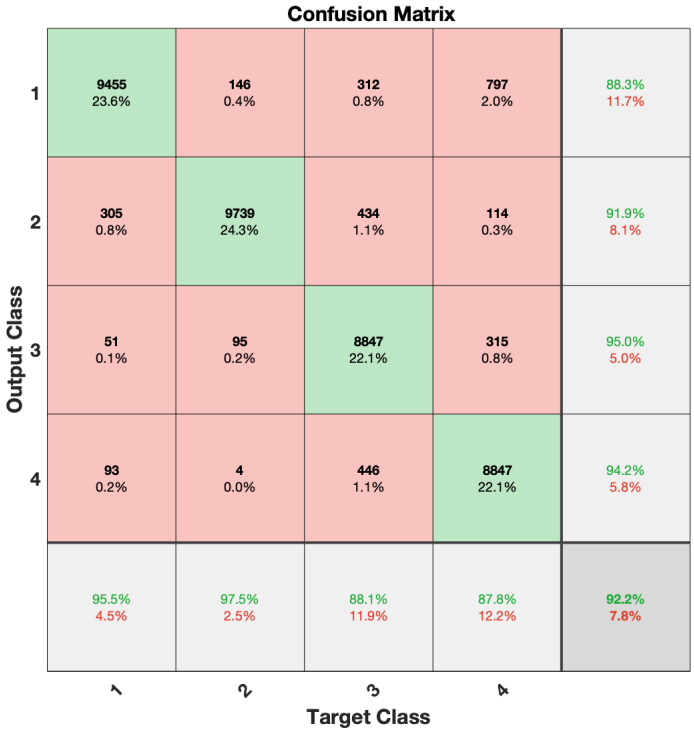


Fig. 7: CNN Confusion Matrix

$p_\theta(z|x)$. Conversely, the encoded distribution is represented by the inference network $q_\phi(z|x)$. The unsupervised data analysis and cluster visualization is performed by plotting directly the inferred mean and logarithm of the variance in a two dimensional space. Indeed, the encoder is comprised of a network that given the input data, predicts the mean and the log of the variance. The network has a common hidden layer comprising of 400 neurons with sigmoid activation functions and two output layers each of additional 200 neurons for prediction of the two distribution parameters. The VAE has been trained using a training set comprising $40,000$ light curves $32,000$ of which are employed for proper training and $8,000$ for network validation. The training occurred for $1,000$ epochs using a batch size of 128 elements. Total time for training is reported to be $2,000$ seconds. As explained above, the VAE learns the latent distribution of the data. By plotting the key parameters of the distribution, i.e. mean and the log of the variance, on a two dimensional plot, structure of the data may emerges. Figure 9 shows the results of the VAE learned parameters for the light curves comprising the training set. Each light curve vector of 721 elements is collapsed into one individual point and colored according the rocket body type class. Points tend to group in four (4) clusters representative of the found rocket body classes comprising the training set. Discrimination between classes is possible as complex decision boundaries can be learned by a highly non-linear classifier such as the CNN.

Next, we contrast the VAE approach based on deep learning, with the t-SNE algorithm. Recently, new probabilistic approaches that preserve the local distance between data while mapping the high-dimensional data into a low-dimensional subspace. One of the most popular approach to such non-linear data dimensionality reduction is the t-Distributed Stochastic Neighbor Embedding (t-SNE, [21]). Similarly to the VAE case, we considered 2-D visualization of the samples light curves employed in the training of the deep networks and we processed the $8,000$ samples (i.e. $2,000$ per class) via the t-SNE algorithm for four selected distances, i.e. mahalanobis, cosine, chebyshev and euclidean. The resulting 2-D embedding is visualized in figures 10(a),10(b),10(c),10(d). Structure of the data and clustering is readily apparent in all cases but the one employing the Mahalanobis distance. All classes seem to be fairly separated in the 2-D embedding which reflects the clustering the 721-dimensional space of the original light curves.

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

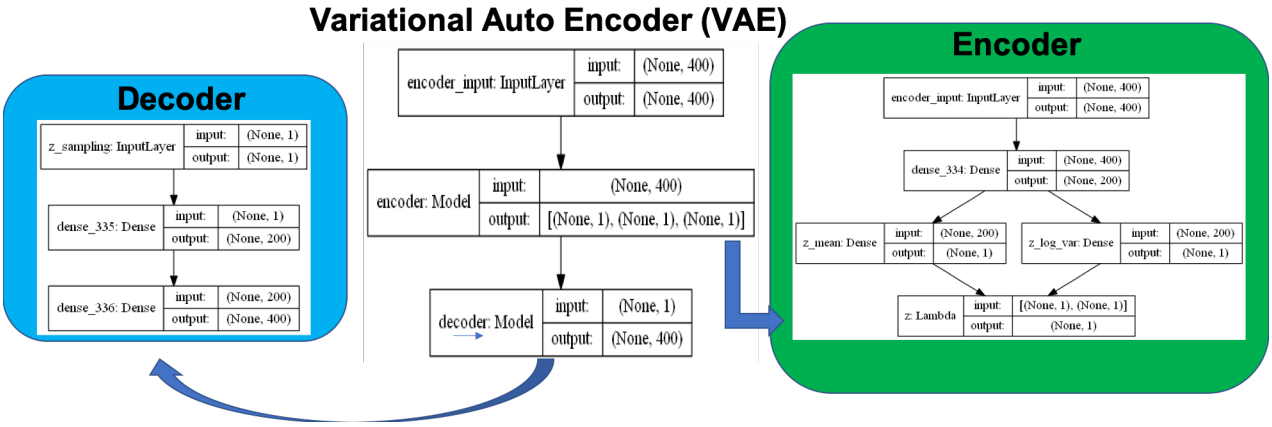


Fig. 8: Variational AutoEncoder Architecture

5. CONCLUSION

In this paper, we explored the feasibility of using a data-driven approach to the solution of the SO light curve inverse problem for SO shape retrieval. Physically-based light curve inversion relies on a first-principles computational model (forward model) to determine the light curve response as function of SO parameters/configuration (e.g. shape). However, the inverse problem is ill-posed and inversion is generally hard. Here, CNNs have been designed and trained to directly learn the inverse functional relationship between measured light curves and shape. More specifically, we considered as set of four classes belonging to the family of rocket bodies and generated a large data set comprising light curves for different SO attitude motion while maintaining a pre-specified orbit. Training and validation show that the CNN is capable of detecting the correct orbit with an accuracy of 92.2% over a validation set comprising $40,000$ samples. Importantly, VAE have been trained in an unsupervised fashion to understand the latent data distribution. Plotting the mean and variance of the training set over a two-dimensional space show the clustering pattern of the data. Importantly, a visual inspection of the cluster structure shows that the decision boundaries between classes are highly complex and interwoven. The results are confirmed by non-linear dimensionality reduction techniques, such as t-SNE, which are capable of projecting high-dimensional data in 2D embedding while presenvind the distance between the samples. As a result, deep networks are demonstrated to be able to solve the inverse problem with good accuracy due to their capability to represent complex non-linear functional relationships.

REFERENCES

[1] Linares, R., Jah, M. K., Leve, F. A., Crassidis, J. L., and Kelecy, T., "Astrometric and Photometric Data Fusion For Inactive Space Object Feature Estimation," *Proceedings of the International Astronautical Federation*, Cape Town, South Africa, Sept. 2011, Paper ID: 11340.

[2] Linares, R., Jah, M. K., and Crassidis, J. L., "Inactive Space Object Shape Estimation via Astrometric And Photometric Data Fusion," *AAS/AIAA Space Flight Mechanics Meeting*, No. AAS Paper 2012-117, Charleston, SC, Jan.-Feb 2012.

[3] Linares, R., Jah, M. K., and Crassidis, J. L., "Space Object Area-To-Mass Ratio Estimation Using Multiple Model Approaches," *Advances in the Astronautical Sciences*, Vol. 144, 2012, pp. 55–72.

[4] Linares, R., Jah, M. K., Crassidis, J. L., and Nebelecky, C. K., "Space Object Shape Characterization and Tracking Using Light Curve and Angles Data," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, 2013, pp. 13–25.

[5] Hinks, J. C., Linares, R., and Crassidis, J. L., "Attitude Observability from Light Curve Measurements," *AIAA Guidance, Navigation, and Control (GNC) Conference*, No. 10.2514/6.2013-5005, AIAA, Boston, MA, August 2013.

[6] Hall, D., Calef, B., Knox, K., Bolden, M., and Kervin, P., "Separating Attitude and Shape Effects for Non-resolved Objects," *The 2007 AMOS Technical Conference Proceedings*, 2007, pp. 464–475.

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

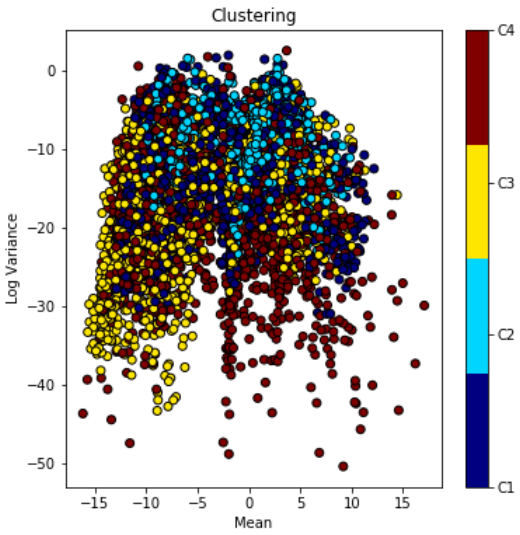
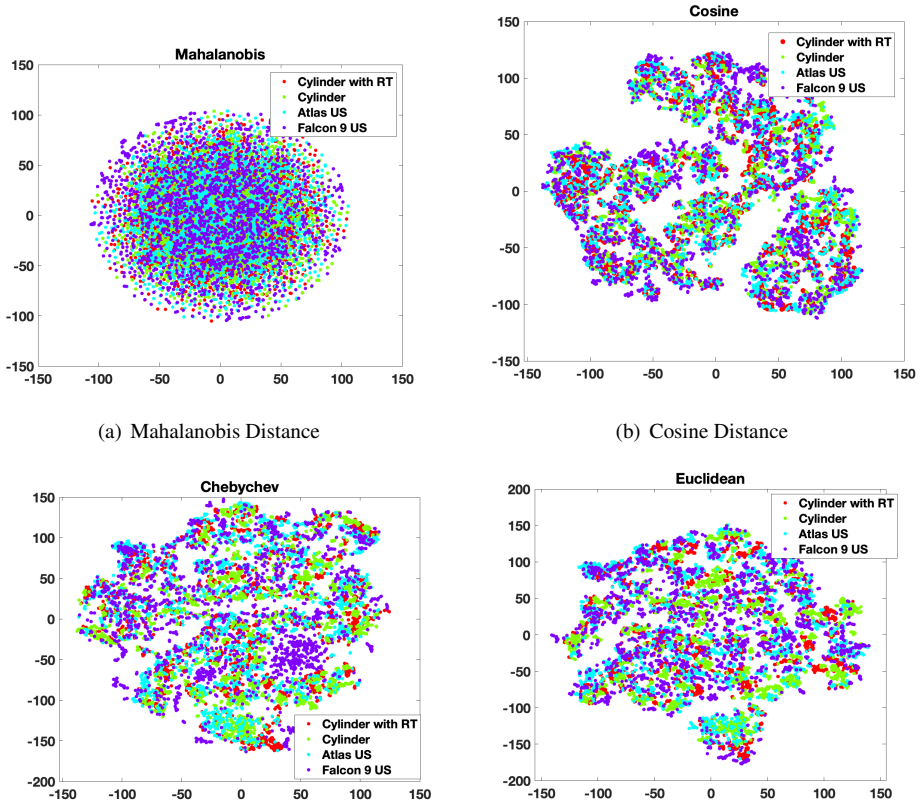


Fig. 9: VAE-learned light curves in the latent variable space



(a) Mahalanobis Distance

(b) Cosine Distance

(c) Chebyshev Distance

(d) Euclidean Distance

Fig. 10: t-SNE algorithms for light curves data distribution in a 2D embedding

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

[7] Linares, R., Shoemaker, M., Walker, A., Mehta, P. M., Palmer, D. M., Thompson, D. C., Koller, J., and Crassidis, J. L., "Photometric Data from Non-Resolved Objects for Space Object Characterization and Improved Atmospheric Modeling," *Advanced Maui Optical and Space Surveillance Technologies Conference*, Vol. 1, 2013, p. 32.

[8] Linares, R., Jah, M. K., Crassidis, J. L., Leve, F. A., and Kelecy, T., "Astrometric and photometric data fusion for inactive space object feature estimation," *Proceedings of 62nd International Astronautical Congress, International Astronautical Federation*, Vol. 3, 2011, pp. 2289–2305.

[9] Gaylor, D. and Anderson, J., "Use of Hierarchical Mixtures of Experts to Detect Resident Space Object Attitude," *Advanced Maui Optical and Space Surveillance Technologies Conference*, Vol. 1, 2014, p. 70.

[10] Wetterer, C. J., Linares, R., Crassidis, J. L., Kelecy, T. M., Ziebart, M. K., Jah, M. K., and Cefola, P. J., "Refining space object radiation pressure modeling with bidirectional reflectance distribution functions," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, 2013, pp. 185–196.

[11] Jah, M. and Madler, R., "Satellite Characterization: Angles and Light Curve Data Fusion for Spacecraft State and Parameter Estimation," *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, Vol. 49, Wailea, Maui, HI, Sept. 2007, Paper E49.

[12] Holzinger, M. J., Alfriend, K. T., Wetterer, C. J., Luu, K. K., Sabol, C., and Hamada, K., "Photometric attitude estimation for agile space objects with shape uncertainty," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 921–932.

[13] Linares, R. and Crassidis, J. L., "Resident Space Object Shape Inversion via Adaptive Hamiltonian Markov Chain Monte Carlo," *AAS/AIAA Space Flight Mechanics Meeting*, No. AAS Paper 2016-514, Napa, CA, Feb 2016.

[14] LeCun, Y., Bengio, Y., and Hinton, G., "Deep learning," *Nature*, Vol. 521, No. 7553, 2015, pp. 436–444.

[15] Lee, H., Pham, P., Largman, Y., and Ng, A. Y., "Unsupervised feature learning for audio classification using convolutional deep belief networks," *Advances in neural information processing systems*, 2009, pp. 1096–1104.

[16] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[17] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A., "Learning deep features for scene recognition using places database," *Advances in neural information processing systems*, 2014, pp. 487–495.

[18] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L., "Large-scale video classification with convolutional neural networks," *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[19] Baldi, P., "Autoencoders, unsupervised learning, and deep architectures," *Proceedings of ICML workshop on unsupervised and transfer learning*, 2012, pp. 37–49.

[20] Doersch, C., "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[21] Maaten, L. v. d. and Hinton, G., "Visualizing data using t-SNE," *Journal of machine learning research*, Vol. 9, No. Nov, 2008, pp. 2579–2605.

[22] Linares, R., Jah, M. K., Crassidis, J. L., and Nebelecky, C. K., "Space object shape characterization and tracking using light curve and angles data," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, 2013, pp. 13–25.

[23] Combal, B., Baret, F., Weiss, M., Trubuil, A., Mace, D., Pragnere, A., Myneni, R., Knyazikhin, Y., and Wang, L., "Retrieval of canopy biophysical variables from bidirectional reflectance: Using prior information to solve the ill-posed inverse problem," *Remote sensing of environment*, Vol. 84, No. 1, 2003, pp. 1–15.

[24] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, Vol. 86, No. 11, 1998, pp. 2278–2324.

[25] Ashikmin, M. and Shirley, P., "An Anisotropic Phong Light Reflection Model," Tech. Rep. UUCS-00-014, University of Utah, Salt Lake City, UT, 2000.

[26] Shuster, M. D., "A Survey of Attitude Representations," *Journal of the Astronautical Sciences*, Vol. 41, No. 4, Oct.-Dec. 1993, pp. 439–517.

[27] Kaasalainen, M. and Torppa, J., "Optimization methods for asteroid lightcurve inversion: I. shape determination," *Icarus*, Vol. 153, No. 1, 2001, pp. 24–36.

[28] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *arXiv preprint arXiv:1603.04467*, 2016.

Copyright © 2019 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com