Prognostics of Combustion Instabilities from Hi-speed Flame Video using A Deep Convolutional Selective Autoencoder

Adedotun Akintayo[1], Kin Gwn Lore[2], Soumalya Sarkar[3] and Soumik Sarkar[4] *

[1,2,4] *Mechanical Engineering Department, Iowa State University, Ames, Iowa, 50011, USA*
akintayo@iastate.edu
kglore@iastate.edu
soumiks@iastate.edu

[3] *Decision Support and Machine Intelligence, United Technologies Research Center, East Hartford, Connecticut, 06118, USA*
sms388@gmail.com

ABSTRACT

The thermo-acoustic instabilities arising in combustion processes cause significant deterioration and safety issues in various human-engineered systems such as land and air based gas turbine engines. The phenomenon is described as self-sustaining and having large amplitude pressure oscillations with varying spatial scales of periodic coherent vortex shedding. Early detection and close monitoring of combustion instability are the keys to extending the remaining useful life (RUL) of any gas turbine engine. However, such impending instability to a stable combustion is extremely difficult to detect only from pressure data due to its sudden (bifurcation-type) nature. Toolchains that are able to detect early instability occurrence have transformative impacts on the safety and performance of modern engines. This paper proposes an end-to-end deep convolutional selective autoencoder approach to capture the rich information in hi-speed flame video for instability prognostics. In this context, an autoencoder is trained to selectively mask stable flame and allow unstable flame image frames. Performance comparison is done with a well-known image processing tool, conditional random field that is trained to be selective as well. In this context, an information-theoretic threshold value is derived. The proposed framework is validated on a set of real data collected from a laboratory scale combustor over varied operating conditions where it is shown to effectively detect subtle instability features as a combustion process makes transition from stable to unstable region.

1. INTRODUCTION

Deep learning models have been shown to outperform all other state-of-the-art machine learning techniques for handling very large dimensional data spaces and learn hierarchical features in order to perform various machine learning tasks. However, most of the studied applications primarily have been in the domains of image, speech and texts processing. For example, convolutional neural network-based applications include object recognition (Farabet, Couprie, Najman, & LeCun, 2013; Akintayo, Lee, et al., 2016), image enhancement (Lore, Akintayo, & Sarkar, 2016), Graph Transformer Networks (GTN) for rapid, online recognition of handwriting (LeCun, Bottou, Bengio, & Haffner, 1998), natural language processing (Collobert & Weston, 2008), large vocabulary continuous speech recognition (Sercu, Puhrsch, Kingsbury, & LeCun, 2016). It is still not common to apply the cutting edge improvements of deep learning towards developing advanced *Prognostics and Health Monitoring* (PHM) algorithm for typical engineering applications. In this paper, we propose a novel selective autoencoder approach within a deep convolutional architecture to analyze hi-speed flame videos for early detection of combustion instability in a gas turbine engine. Whereas traditional PHM algorithms mainly use time series data (e.g., pressure and temperature etc.). For this purpose, the proposed approach attempts to advance PHM via capturing the rich information of hi-frequency video. The approach performs implicit labeling in order to derive soft labels from extreme classes that are explicitly labeled as either positive or negative examples. This particular property is significant for tracking continuous temporal phenomenon such as the transition from combustion stability to instability, where labels of extreme states (stable or unstable) are available but intermediate state labels are not. Explicit labels are utilized to selectively mask selective features while allowing other features to remain. Fig. 1 shows grayscale im-

ages describing typical gradual development of instability at the stated parameters in the swirl-stabilized combustor used for the experiment.
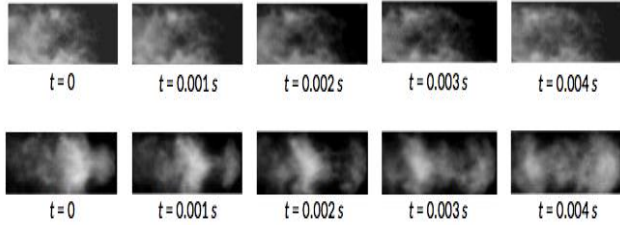

Figure 1. Grayscale images of gradual time-varying development of instability structure at two different parameter values

Labeling (e.g., structured and implicit) can be considered a multi-class classification problem (Erdogan, 2010). For example, three-stage Hidden Markov Models (HMM) were used for handling speech recognition (Rabiner, 1989) problems, parts of speech tagging (Meyer, 2011-2012) and sequence labeling because they derive the relationships from observations-to-state and state-to-state in dynamic systems. Maximum Entropy Markov Model (MEMM), a discriminative modification of HMM, was introduced to overcome the latter's recall and precision problems especially in labeling texts. In those models, conditional probability of the desired labels are learnt directly based on the uncertainty maximization idea. Applications of MEMM for natural language processing can be found in (Berger, Pietra, & Pietra, 1996).

Due to "label bias" defects of MEMM, a Conditional Random Field (CRF), which is a joint Markov Random Field (MRF) of the states conditioned on the whole observations is later explored (Lafferty, McCallum, & Pereira, 2001). It enabled considering the global labels of the observation as against localization of labels of MEMM (Erdogan, 2010). However, labeling in this case is made computationally complex by the relaxation of statistical independence assumption of the observations which most of the models assume.

Recurrent Neural Networks (RNNs) have been utilized for sequence labeling problems due to its cyclic connections of neurons (Graves, 2014) as well as its temporal modeling ability. Although earlier construction of RNNs is known to have short ranged memory issues and a restrictive unidirectional information context access, formulation of a bidirectional Long Short Term Memory (LSTM) (Graves & Schmidhuber, 2005) resolved such issues. However, this construction adds to the complexity of the model significantly as typically two RNNs get connected through the same output layer.

From the application standpoint, early detection of instability in the combustion chambers of dynamic systems aids anticipative actions for reducing its consequent effects. Visual-

izing the features that characterizes the intermediate frames of its spectrum is an important approach to unravel the processes that precede instability. The authors in (Sarkar, Lore, Sarkar, Ramaman, et al., 2015) introduced Deep Belief Networks (DBN) as a viable technique to achieve the aim with a view to exploring other machine learning techniques for confirmation. They improved on that by applying a modular neural-symbolic approach (Sarkar, Lore, & Sarkar, 2015) in another publication.

In this paper, we propose a deep convolutional selective autoencoder-based anomaly (early) detection framework for the crucial physical process of combustion for better understanding of the underlying complex physics. Combustion instability is a significant anomaly characterized by high-amplitude flame oscillations at discrete frequencies that reduces the efficiency and longevity of aircraft gas-turbine engines. Full-blown instability can be differentiated from stable combustion via video analysis with high confidence because unstable combustion flames show distinct coherent structures similar to 'mushroom' shapes. But it is extremely difficult to detect an onset of instability early due to fast spatio-temporal transience in the video data. Therefore, the instability detection problem boils down to an implicit soft labeling problem where we train a deep model using hi-speed flame videos with explicit labels of stable and unstable flames such that it recognizes the onset of instability early as the combustion process makes transition from a stable to unstable region.

Conceptually, this is similar to cognitive psychologists' description of human reasoning in object classification (Tenenbaum, Kemp, Griffiths, & Goodman, 2011). An example is to consider how a child is taught on intrinsic classes. A similar problem is how to detect a cross breed of dog and wolf and how close the animal is to either of the classes. From an application standpoint, an early detection of engine's combustion instability may be useful for computing the instantaneous values of the remaining useful life, but the computation is partial since other engine physical use factors are also important. Therefore, remaining useful life (RUL) computation is beyond the scope of the present problem.

Contributions: The main contributions of this paper is delineated below:

- A convolutional selective autoencoder framework based on emerging deep learning techniques is proposed for a significant PHM application - early detection of combustion instability;

- The method avoids extensive expert-guided feature hand-crafting (Farabet et al., 2013) while addressing a complex physical phenomenon like combustion to discover coherent structures in flames images;

- The proposed framework is able to learn from high dimensional data sets (e.g., high speed video) of most applications and provides a platform for determining the

degree of relationship between the states of two temporally close observations;

- A metric to desired level of granularity is constructed to track the onset of combustion instability and detect pre-transition phenomena such as 'intermittence'. Intermittence is a temporary (of the order of millisecond, equivalent in this case to few video frames) blast of instability characterized by small and partially observable coherent structure;

- Extensive validation and comparison using CRF technique are provided based on laboratory-scale combustion data collected under various realistic operating conditions.

Paper organization: The paper is organized in six sections including the present one. Section 2 presents prior work on the proposed approach and problem formulation. In section 3, the main architecture for the problem is discussed followed by a distance metric that is used to access the result quantitatively. Section 4 provides an opportunity to introduce the problem dataset collection and then the implementation of the composite architecture as well as the competing method. The results obtained for the hypothesis is discussed in section 5. We conclude the paper in section 6 as well as give some insights into the direction of future works.

2. BACKGROUND

This section provides a brief overview of convolutional networks, a description of the example problem of detecting combustion instability, and the notion of implicit labeling.

2.1. Convolutional networks

Convolutional networks (Krizhevsky, Sutskever, & Hinton, 2012) are a type of deep networks that offer discriminative advantages as in the MEMM as well as providing global relationship between observations as in the CRF. The architectures rely primarily on local neighborhood matching for data dimension reduction using nonlinear mapping (i.e., sigmoid, softmax, hyperbolic tangent and ReLU). Each unit of the feature maps has common shared weights or kernels for efficient training with relatively–compared to fully connected layers– lower trainable parameters. Feature extraction and classifier learning are the two main functions of these networks (LeCun et al., 1998). However, to learn the most expressive features, we have to determine the invariance-rich codes embedded in the raw data and then, a fully connected layer to reduce further the dimensionality of the data and map the most important codes to a low dimension of the examples. Many image processing and complex simulations depend on the invariance property of the convolution neural network stated in (LeCun & Bengio, 1998) to prevent overfitting by learning expressive codes.

The feature maps are able to preserve local neighborhood pat-

terns for each receptive field as with over-completeness dictionary in (Aharon, Elad, & Bruckstein, 2006). A full and detailed review may be found in (LeCun et al., 1998) where the authors note the advantage of local correlation enforcing convolution before spatio-temporal recognition. For efficient learning purposes, convolutional networks are able to explore the benefits of distributed map-reduce frameworks (Fung & Mann, 2004) to leverage large training data as well as multi-GPU computing. With these benefits, the winners of the ILSVRC 2012 (Krizhevsky et al., 2012) utilized a large network of 8 layers and 2 GPUs training on the same architecture provided in (LeCun et al., 1998) to achieve the then best position. Subsequently, GoogLeNet (Szegedy et al., 2015) and other authors (Simonyan & Zisserman, 2015) have also reported better performance with larger models found to be more related to the depth of the network.

2.2. The problem of combustion instability

Combustion instability reduces the efficiency and longevity of aircraft gas-turbine engines. It is considered a significant anomaly characterized by high-amplitude flame oscillations at discrete frequencies. These frequencies typically represent the natural acoustic modes of the combustor. Combustion instability arises from a positive coupling between the heat release rate oscillations and the pressure oscillations. Coherent structures are fluid mechanical structures associated with coherent phase of vorticity (Hussain, 1983). The generation mechanisms of the structures vary system wise, causing large scale velocity oscillations and overall flame shape oscillations by curling and stretching. These structures can be caused to shed–or be generated–at the duct acoustic modes when the forcing (pressure) amplitudes are high. There is a lot of recent research interest on detection and correlation of these coherent structures to heat release rate and unsteady pressure. The popular methods resorted for detection of coherent structures are proper orthogonal decomposition (POD) (Berkooz, Holmes, & Lumley, 1993) (similar to principal component analysis (Bishop, 2006)) and dynamic mode decomposition (DMD) (Schmid, 2010), which use tools from spectral theory to derive spatial coherent structure modes.

2.3. Implicit labeling

Semi-supervised training for classification takes advantage of the labels at the final layers. A variant of structured labeling by (Kulesza, Amershi, Caruana, Fisher, & Charles, 2014) called implicit labeling is used to derive soft labels from extreme classes that are explicitly labeled as either positive or negative examples. Explicit labels usually can be utilized to selectively mask one feature, especially that one is not interested in while parsing the class of interest. However, explicit labels on its own can only serve as a classifier for intrinsic classes in the test sets learnt from the training set. Implicit labeling here also bears similarity to the sequence

Figure 2. Illustration of implicit method of generating soft labels

labeling (Erdogan, 2010) with an extra constraint of utilizing prior knowledge provided only by explicit label. It is then fused with convolutional auto-encoder architecture algorithm described in section 3.1 to determine the intermediate or transition phases–a mixed breed of a dog and a wolf for instance– and more importantly to what degree is the animal a dog or a wolf. Thus, it attempts to derive soft labels from expert-informed, hard-mined labels as illustrated in fig. 2 with a composite architecture.

3. ALGORITHMS

In this section, the algorithms for sequence labeling are described. We provide a little more details of the convolutional autoencoder and its interface with the selectivity criterion. Subsequently, a brief background on the conditional random field (CRF) algorithm is provided. Then, we discuss the information theoretic metrics that facilitate image dimensionality reduction, and the basis for our threshold computation.

3.1. Convolutional Selective Autoencoder

Based on the convolutional network's (*convnet* for short) performances on several similar tasks reviewed, it is found a suitable candidate for the composite architecture to examine our hypothesis of soft label generation. A convnet architecture for low-level feature extraction with a symbolic graphical model such as STSA at the top level (Sarkar, Lore, & Sarkar, 2015) has been previously used for this problem. In contrast, we use an end-to-end convolutional selective auto-encoder (as shown in fig. 3), designed and tested (Akintayo, Lee, et al., 2016) to explore another perspective to the current problem. The constituent steps for the model to learning from the data are outlined below.

Explicit labels and pre-processing: Given an $M \times N$ dimensional image frames and corresponding ground truth labels (one of the two classes), explicit labels are generated by selectively masking frames with the undesired class with black pixels. Hence, N pairs of input-output pairs $\{(X_i, Y_i)\}$ for $i = 1, 2, ..., N$ are generated where X represents the original images, Y are the masked frames that are considered explicitly as ground truth. The images are then normalized where pixel intensities have zero mean and a standard deviation of 1 as preprocessing.

Convolutional layers: Convolutional autoencoders (CAE), also called deconvolution nets (Zeiler & Fergus, 2014) or fully convolutional networks (Long, Shelhamer, & Darrell, 2015) start with propagation from the input layer to the convolution layer. Also, the step before the output layer is the deconvolution layer. At each convolution or deconvolution layer, a chosen $(c \times c)$ filter size is convolved with the patches to learn a z_o−dimensional feature map from which joint weight over the z_i−dimensional feature maps that are useful for enforcing local correlation is learnt to characterize all maps as,

$$\hat{Y}_{z_o(m-c+1)(n-c+1)} = C[X_{z_i mn} \star W_{z_i cc} + b_c] \qquad (1)$$

where C is the squashing function, $\star$ is the convolution operator of the joint weights, $W_{z_i cc}$, b_c the biases and input from previous layer $X_{z_i mn}$. To enhance the invariance further, pooling is done to propagate representative features in local neighborhoods. It ensures that all the neurons activation in a locality do not have high entropy enough such that information is diffused. In this case, maxpooling (Scherer, Muller, & Behnke, 2010) is selected as a representative for a $p \times p$ neighborhood.

Fully connected layers: The feature maps from the previous convolution and subsampling layers are flattened. In order to reduce the number of parameters for the fully connected layers, combat the problem of overfitting and to avoid getting trapped in local optima, some features are randomly left out with a dropout layers (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012). Dropout in the hidden layer produces better results as it eliminates the necessity for regularization parameters used previously (Akintayo, Lore, Sarkar, & Sarkar, 2016). A layer encodes the most important feature from the input of the previous layer with $\hat{Y}_e = E[W_e \hat{Y} + b_e]$ and another layer reconstruct the useful features with $\hat{Y}_d = D[W_d \hat{Y}_e + b_d]$, where E and D stands for the rectified linear unit (ReLU)-type encoder and decoder functions respectively. b denotes the biases and W denotes the weights of the layer. The subscripts e and d indicates the encoder and decoder. Note that the ReLU nonlinearity on a parameter is represented by $\text{ReLU}(f) = \max(0, f)$. Intuitively, it has the advantage of easier training compared to other nonlinearity types because the activations of each neuron is a piece-wise linear function of argument f and do not saturate.

Unpooling: In this layer, a reversal of the pooled dimension is done by stretching and widening (Jones, 2015) the identified features from the filters of the previous layer. It is also an upscaling of the feature maps around the axes of symmetry where the reconstructed feature maps are optimized through the back-propagation algorithm.

Error minimization: This phase is akin to a feedback stage
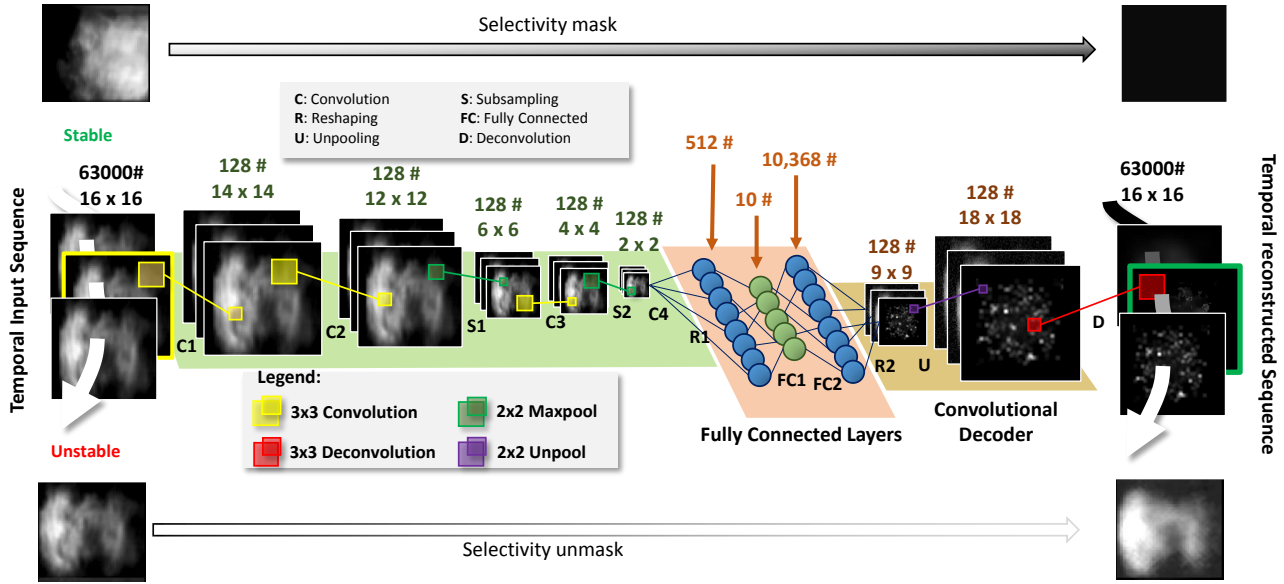
Figure 3. Structure of the convolutional autoencoder with selectivity masks. The encoder portion extracts meaningful features from convolution and sub-sampling operations, while the decoder portion reconstructs the output into the original dimensions through deconvolution and upsampling. Best viewed on screen, in color.

in a control paradigm or a scenario where a teacher–labeled data–provides feedback in performance measure on how well a student–the machine–has learned features related to a particular application–the task. The process included a regularization function as in (LeCun et al., 1998) to avoid overfitting the data. The Nesterov momentum-based (Sutskever, Martens, Dahl, & Hinton, 2013) stochastic gradient descent is used for improved results when compared to other loss functions such as adaptive subgradient (ADAGRAD) (Duchi, Hazan, & Singer, 2011) and adaptive learning rate method (ADADELTA) (Zeiler, 2012) for the reconstruction error updates given the reconstructed output $\hat{Y}_{mn}$ and the labels, Y_{mn}. Let $\theta = \{\mathbf{W}, \mathbf{b}\}$ be the set of weights and biases for all layers that are to be optimized by minimizing the loss function $L(\theta)$. The loss function is a mean square error cost function given by,

$$L_{train}(\theta) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (Y_{ij} - \hat{Y}_{ij})^2 \qquad (2)$$

Subsequently, the weights are updated at each time step, k, via stochastic gradient descent (LeCun et al., 1998),

$$W^{(k)} = W^{(k-1)} - \alpha \frac{\partial L(W^{(k-1)})}{\partial W^{(k-1)}} \qquad (3)$$

where α is the learning rate equivalent of step size in optimization problems. More details can be found in (Masci, Meier, Ciresan, & Schmidhuber, 2011) while the background materials presented thus far and those in subsection 3.3 are the more important aspects to describe our embedded improvements.

3.2. Conditional Random Field (CRF)

CRF is another class of well-studied (Domke, 2013) and formulated models for labeling problems. It is an improvement of the Markov Random Field, MRF where one is interested in determining the conditional probabilities of newer observation such as our test data given the knowledge of previous ones such as the explicit labels. The benefits of CRFs are their improvements in the learning stage on previous likelihood estimation by including inference approximation. The algorithms have been shown (Barbu, 2009) to perform well on complex image problems such as image denoising task as well as being robust to model misspecification. Therefore, we also incorporated selectivity condition into the CRF in a similar way to that of CAE.

3.3. Instability Metric

Similar to that presented in (Liu, Ghosal, Jiang, & Sarkar, 2016), a metric based on the Kullback-Liebler (KL) divergence (Kullback & Liebler, 1951) is chosen to measure the distance of the results from the image frames in each transition protocol from the expected result of a stable flame frame. This yields a KL distance, z for each image frame, $\mathbf{I} \in \mathcal{I}$, where $\mathcal{I}$ represents the set of input images frames. It can be expressed mathematically as,

$$z(\mathbf{I}) = \sum_{i \in \mathbf{I}} \lim_{\mathbf{T}(i) \to 0^+} \mathbf{I}(i) log \frac{\mathbf{I}(i)}{\mathbf{T}(i)} \qquad (4)$$

where i represents each pixel in the image frame and $\mathbf{T}$ represents the training label/target image. The implication of the limit is that we intend to drive the flame image pixel values

to zero in the stable combustion region. This physically corresponds to taking the distance of each image from the reference of the stable flame. The present metric has the advantage of using a common reference for all the test transition protocols rather than being specific to a particular image frame within one test protocol (Akintayo, Lore, et al., 2016).

4. DATASET AND IMPLEMENTATION

In this section, we motivate the attempts at solving the problem by describing the dataset, the experimental setup for gathering the data and how it is collected. We also describe the implementation of the two competing algorithms by explaining the choices that were made and stating the important selected parameters for such choices. Finally, the threshold values for analyzing the results are determined.

4.1. Dataset collection and experimental setup

To collect training data for learning coherent structures, thermoacoustic instability was induced in a laboratory-scale combustor with a 30 mm swirler (60 degree vane angles with geometric swirl number of 1.28). Fig. 4 (a) shows the setup and a detail description can be found in (Sarkar, Lore, Sarkar, Ramaman, et al., 2015). In the combustor, 4 different instability conditions are induced: 3 seconds of hi-speed videos (i.e., 9000 frames) were captured at 45 lpm (liters per minute) FFR (fuel flow rate) and 900 lpm AFR (air flow rate), and at 28 lpm FFR and 600 lpm AFR for both levels of premixing. Fig. 4 (b) presents sequences of images of dimension 100×237 pixels for unstable ($AFR = 900lpm$, $FFR = 45lpm$ and full premixing) state. The flame inlet is on the right side of each image and the flame flows downstream to the left. As the combustion is unstable, fig. 4 (b) shows formation of mushroom-shaped vortex (coherent structure) at $t = 0, 0.001s$ and the shedding of that towards downstream from $t = 0.002s$ to $t = 0.004s$. For testing the proposed architecture, 5 transition videos of 7 seconds length were collected where stable combustion progressively becomes unstable via 'intermittence' phenomenon (fast switching between stability and instability as a precursor to persistent instability) by reducing FFR or increasing AFR. The transition conditions are as follows (all units are lpm): (i) AFR = 500 and FFR = 40 to 28, (ii) AFR = 500 and FFR = 40 to 30, (iii) FFR = 40 and AFR = 500 to 600, (iv) AFR = 600 and FFR = 50 to 35, (v) FFR = 50 and AFR = 700 to 800. For clarity, these data sets are named as 500_{40to38}, 500_{40to30}, $40_{500to600}$, 600_{50to35}, and $50_{700to800}$ respectively for analysis in the subsequent sections of this paper.

4.2. Training process

In training the networks, $63,000$ grayscale frames having dimensions 100×237 are resized to 16×16 for computa-

tional simplicity. A total of $35,000$ frames is labeled stable while the remaining $28,000$ were labeled unstable. These images were a combination of datasets with different premixing lengths of either 90mm or 120mm and a wide range of air and fuel LPMs for which the combustor is either in a stable or an unstable state. The whole training dataset is divided into two parts: 75% of it is used to train the algorithm, while 25% is held out for validating their results and setting our thresholds.

CAE: The convolutional autoencoder parameters include learning rate of 0.0001 with momentum = 0.975 is found to train the model best in the Nesterov based stochastic gradient descent formulation. The network is trained to 100 epochs in order to conveniently strike a good minima of the validation error. Training is done on GPU Titan Black with 2880 CUDA cores, equipped with 16MB video RAM, using the python-based machine learning frameworks such as Theano, Lasagne and NoLearn (Bergstra et al., 2010; Thoma, 2016). Lasagne offers a wide variety of control over the layer types, nonlinearity types, objective functions, interfacing with The- ano, and many other features built into it. NoLearn, on the other hand, is a coordinating library for the implementation of the layers in Lasagne which offers model visualization features. While training, a filter of $c \times c$ pixels ($c = 3$ in the implementation) and a non-overlapping $p \times p$ ($p = 2$) maxpooling were found to be experimentally less costly to produce the results. Algorithm training is done in batches of 128 training examples which is found to be suitable via cross validation. The
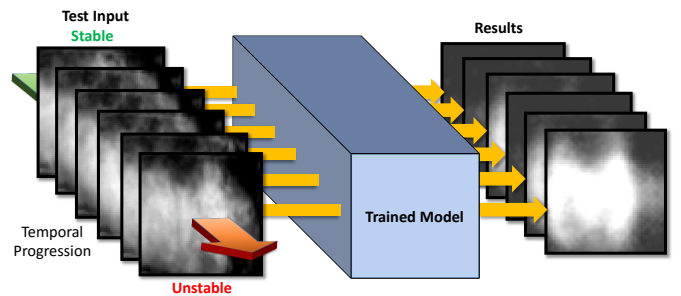

Figure 5. Schematics of implementation of trained network on transition test data

architecture in fig. 3 shows how the layers are interlinked in the training stage which leads to an overall of $416,779$ learnable parameters. From this point onwards, CAE model that is trained to be selective is referred to as convolutional selective autoencoder (CSAE).

CRF: In training the linear to linear type conditional random field, the main hyperparameters are again the loss function which usually is approximated and how the gradient of such objective function are computed. For the present problem, based on multiple trials for hyperparameter, we found the loopy variant of the truncated tree re-weighted (TRW) belief propagation a good inference type for the problem.

(a)

$t = 0$ s $t = 0.001$ s $t = 0.002$ s

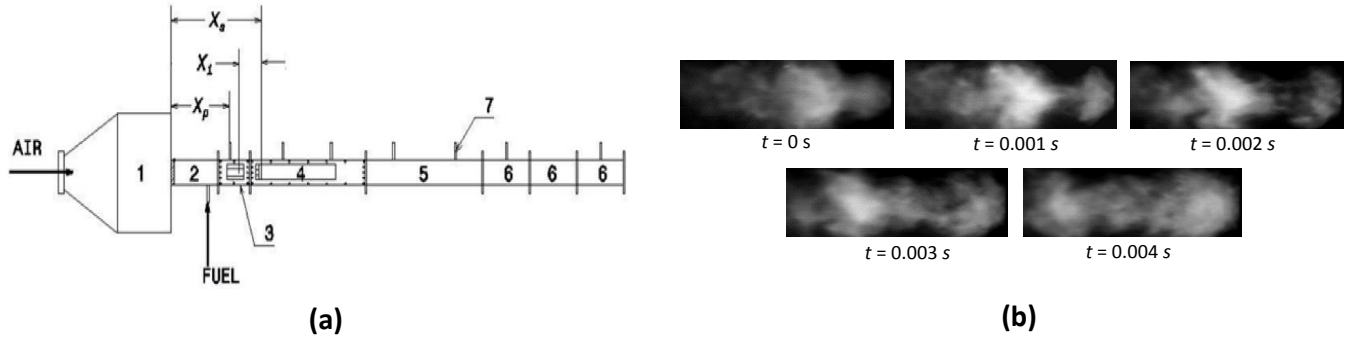$t = 0.003$ s $t = 0.004$ s

(b)

Figure 4. a) Schematics of the experimental apparatus. 1 - settling chamber, 2 - inlet duct, 3 - inlet optical access module (IOAM), 4 - test section, 5 & 6 - big and small extension ducts, 7 - pressure transducers, X_s - swirler location, X_p - transducer port location, X_i - fuel injection location, (b) Visible coherent structure in grayscale images at 900 lpm AFR and full premixing for 45 lpm FFR

Furthermore, for better performance, we chose a clique type loss because of the benefits over simple univariate type loss. A quasi-netwon method, Broyden-Fletcher-Goldfarb-Shanno (BFGS) was chosen to optimize its error backpropagation. The algorithm is also implemented in batches of 512 to reduce computation time, and in a gradual fashion while the regularization parameter used was 0.0001. The model resulted in 8064 cliques. Subsequently, like the CAE, we refer to a CRF model that is trained to be selective as selective conditional random field (SCRF).

4.3. Threshold Determination

Given the models learnt from each of the algorithms, CSAE and SCRF individually, with the training sets as illustrated in fig. 6, the algorithms are separately validated on the validation set. The validation result for each algorithm is used to determine the value of the instability metric, z at which transition takes place, called transition threshold. This is taken as the upper limit of the 95% confidence interval (CI) for the distribution of z (see eqn. 4) for stable flame frames. The schematics in fig. 6 summarizes how it is implemented for each algorithm. We note that this helps to utilize expert knowledge regarding the stable and unstable regions to determine the start 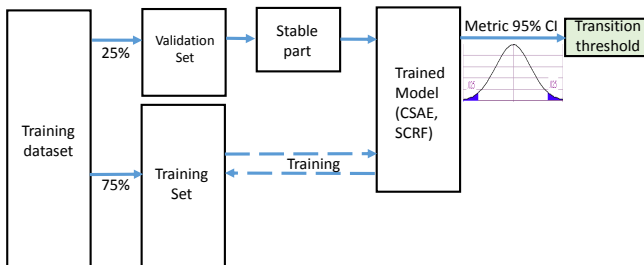of transition from the stable region. Note that these are derived by replacing $\mathcal{I}$ in subsection 3.3 with the known stable part in the validation results.

5. RESULTS AND DISCUSSIONS

In this section, results obtained from the algorithms are discussed and analyzed. The subsections are arranged to build up the argument for early detection of unstable region's properties in frames. Such unstable flame properties can be detected even in the transition region enabling early instability detection. Then we discuss how the network explores the space between the stable and unstable regions to get softer labels. Let the stable region be denoted by 'SR' on one end of the spectrum and the unstable region be 'UR' on the other end of the spectrum. Note, training of the algorithm is performed with explicitly available ground truth labels. The ground truth labels are categorized into frames of stable flame types and frames of unstable flame types. As discussed before, units of frames in the stable region are masked with '0', while those in the unstable region are retained during training. Figure 7 shows the algorithm's ability to satisfy the training criteria in one stable and one unstable validation frames. Figure 7 shows how CSAE learns to be selective in masking the stable region as trained. Feature maps from the model are shown in fig. 8 to highlight the detected features and the reconstructed outputs. For frames closer to UR in the transition stage, the corresponding feature maps showed more pixels activated mushroom structures that characterize UR. For frames in SR however, information is seen to be rapidly diffusing from the input into the hidden layers. At each layer, joint parameters capture the trade-off between discarded and retained information from the stable and unstable training sets. The fully connected layers serve at least two important purposes, namely: (1) to reduce further the image dimensions towards only rich explanatory features, and (2) ensuring structural consistency for optimal layer-wise features by reshaping the output images into dimensions similar to the input. Due to the importance of the layer, an optimal number of units search is re-

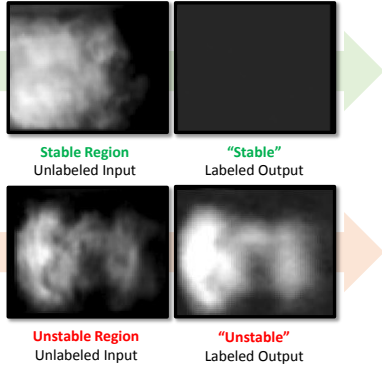Figure 6. Schematics of selection of transition threshold

Figure 7. Illustration of CSAE's ability to reproduce explicit labels.
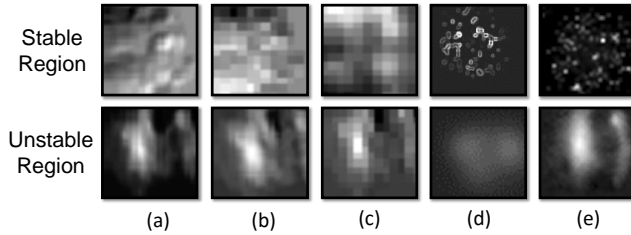

Figure 8. Feature maps for (a) the third convolution layer, (b) the second pooling layer, (c) the fourth convolution layer, (d) the unpooling layer and (e) the deconvolution layer.

ported in the next subsection.

5.1. Optimal Code layer size

Among the many models parameters, the main influencing parameters that motivated this search is the size of the encode layer of the CAE. This is also related to the number of output values of the CRF model. Having the speed-up provided by the GPUs for training CSAE, a search for an optimal size of the code layer is conducted. It is done to reduce arbitrariness in the choice of the number of coding units, and to ensure obtaining the most effective results. Therefore, 100 epochs of CSAE algorithm is run for each of code layer sizes: 8, 10, 20 and 40 units. We started off with 8 units because of its closeness to the presence of two classes in the training data. Then, we allowed more degrees of freedom to see which result demonstrated mostly, the known physical properties of short time bursts while achieving the goals for our training, i.e., selectivity. The results in fig. 9 and every other results in the following subsections are also uniformly smoothed with a simple locally weighted moving average filter Matlab function `loess` having a span of 0.1 to arrive at the smoothed lines. Transition threshold described in subsection 4.3 are shown on each plot of fig. 9. The transition thresholds with

respect to 8, 10, 20 and 40 units at the coding layer are found to be 0.003455, 0.003901, 0.003438 and 0.005738 respectively.

The results in fig. 9 corroborate our previous results (Akintayo, Lore, et al., 2016; Sarkar, Lore, & Sarkar, 2015; Sarkar, Lore, Sarkar, Ramaman, et al., 2015) of transition stage being between the two regions. It is observed that with 40 units, algorithm does not satisfy the selectivity condition of masking the stable part unlike the other units. This may happen due to the decrease in noise rejection capability with increase in degrees of freedom at the coding layer. Also, the discriminatory ability of the results are assessed. It is a metric that quantifies the maximization of the inter-region separation, while minimizing the intra-region separation similar to a Fisher Linear Discriminant analysis. However, for result assessment in this problem, a conservative way is to examine ratio of the variance to the mean provided. The larger the spread around the average, the more the discrimination capability between stable and unstable regions. Therefore, the distribution of z found in eqn 4 are also examined on this basis for each of the test protocols.

From the trends of the statistics on table 1, including early signal of the transition shown by the frame #, coding layer with 10 units produced the best results, both visually and statistically. It however fails to be the most discriminatory due to its large mean despite also having the largest variance. We note that performance improves with increase in coding layer length from 8 to 10, while it reduces when the coding layer length is increased further. While an optimal length of the coding layer can be found between 10 and 20, we selected 10 units for performance comparison with SCRF presented in this paper. Transition frame # for 40 units of the layer is not easily found because the validation results are less suppressed compared to the test frame. Hence, in this case early detection may not be feasible.

Table 1. CSAE optimum encode layer size metric and transition start frame # for protocol 500_{40to30}

# of units	$\mu(\mathbf{z})$	$\Sigma(\mathbf{z})$	$\frac{\Sigma(z)}{\mu(z)}$	frame #
8	0.0222	0.0238	1.071	11710
10	0.0289	0.0302	1.045	11650
20	0.0241	0.0258	1.072	11710
40	0.0175	0.0246	1.041	≈ 12620

5.2. CSAE and SCRF comparison

A visual comparison of the distributions of z (eqn 4) on test transition protocol, 500_{40to30} via their instability metrics are plotted against frame number for both algorithms. These are shown in fig. 10. Clearly, the results of CSAE is more discriminatory in nature, i.e., it has more scatter around its local mean than that of SCRF. CSAE also shows a greater capability than SCRF, to satisfy the training criteria on a new test
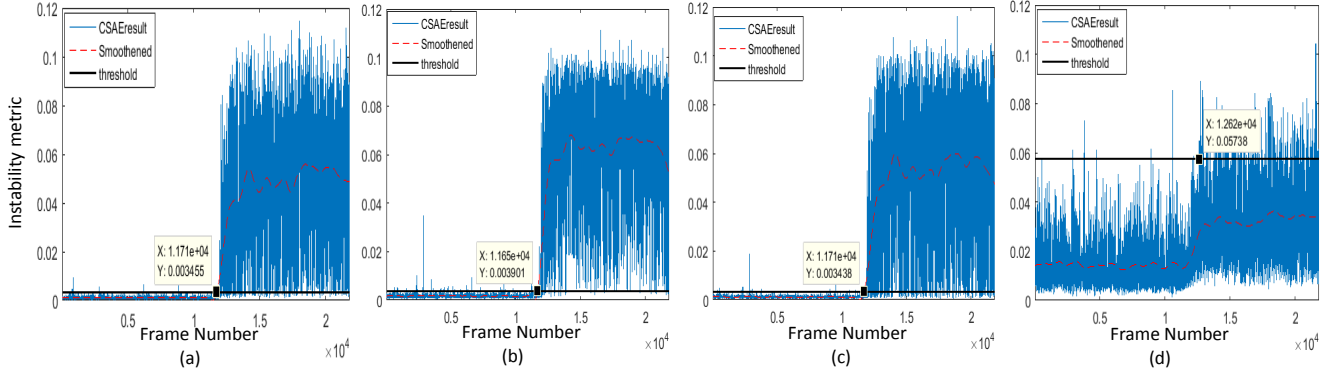
Figure 9. Code layer selection for 500_{40to30} with, (a). 8 units, (b). 10 units, (c). 30 units and (d). 40 units

data set. Therefore, CSAE will be more effective for early detection of instability. Note, the transition threshold for SCRF as defined in subsection 4.3 is found to be 0.03636. On the other hand, threshold for CSAE with 10 code layer units is 0.003901.
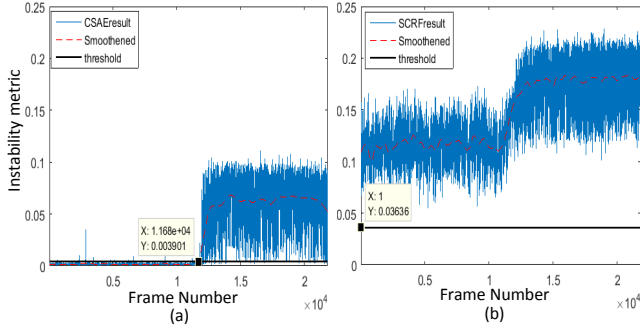

Figure 10. Results of: (a). CSAE and (b). SCRF for test transition protocol, 500_{40to30}

Other differences in computation and memory complexity are shown in table 2. Note that with the demonstrated advantage

Table 2. CSAE and SCRF comparison

Factor	CSAE	SCRF
platform	Python	Matlab
model size	$\mathcal{O}$(Mbytes)	$\mathcal{O}$(Kbytes)
CPU inference rate	$\approx 4.7ms/frame$	$\approx 0.24secs/frame$
GPU inference rate	$\approx 1\mu s/frame$	–

of GPUs, dedicated field programmable gate arrays (FPGAs) built for the proposed CSAE scheme can enable an on-line real-time instability detection tool for real engines.

5.3. Early detection

The speed of detection is in terms of the number of frames seen in the stable region before bursts of instability are detected. However, due to the consistency of the CSAE algorithm with our selective training and domain knowledge (i.e., most of the stable frames are suppressed) on the problems analyzed, its results for 4 test transition protocols are shown and discussed in this subsection. CSAE results on different test transition conditions are presented in fig. 11. It shows the capability of the model to suppress stability features of frames in the SR, while revealing some anomalous instability features in the same frames. It also shows the anomalies to be more prominent in the transition regions. Instability metric introduced in section 3.3 has been used to evaluate the strength of each algorithm's ability to mask examples closer to the SR compared to those nearer to the UR. The results are comparable with those found in (Sarkar, Lore, & Sarkar, 2015) where the framework used a neural-symbolic approach with a combination of convolutional neural networks and symbolic time series analysis to obtain instability metrics. Note, no background knowledge is provided other than domain knowledge regarding the possibility of short-time instability bursts in the stable regions. Figure 11(a) and (b) have similar transition conditions. The latter has a leaner mixture and it shows more short term fluctuations in the post-transition phase compared to (a) (as marked by a dotted box in (b)). Furthermore, it signals earlier (at frame 42) regarding the presence of instability compared to (a) where first indication is approximately around frame 2870. Moreover, possibly in accordance with what is known from physics (Li, Zhou, Jeffries, & Hanson, 2007) about lean mixtures, the protocol in (c) has the most unstable intermittency in both the SR and the transition phase. It may be considered to be the closest to instability of all the protocols as highlighted in the example frames. In contrast to (c), transition protocol in (d) generally shows results that are closer to stability. This is probably due to the balance provided by its originally richer mixture. It also has the most 'late detection' of the early burst of instability as well as departure from stability among all the protocols.

Finally, table 3 shows a summary of the results obtained from the algorithms for all the test transition protocols.
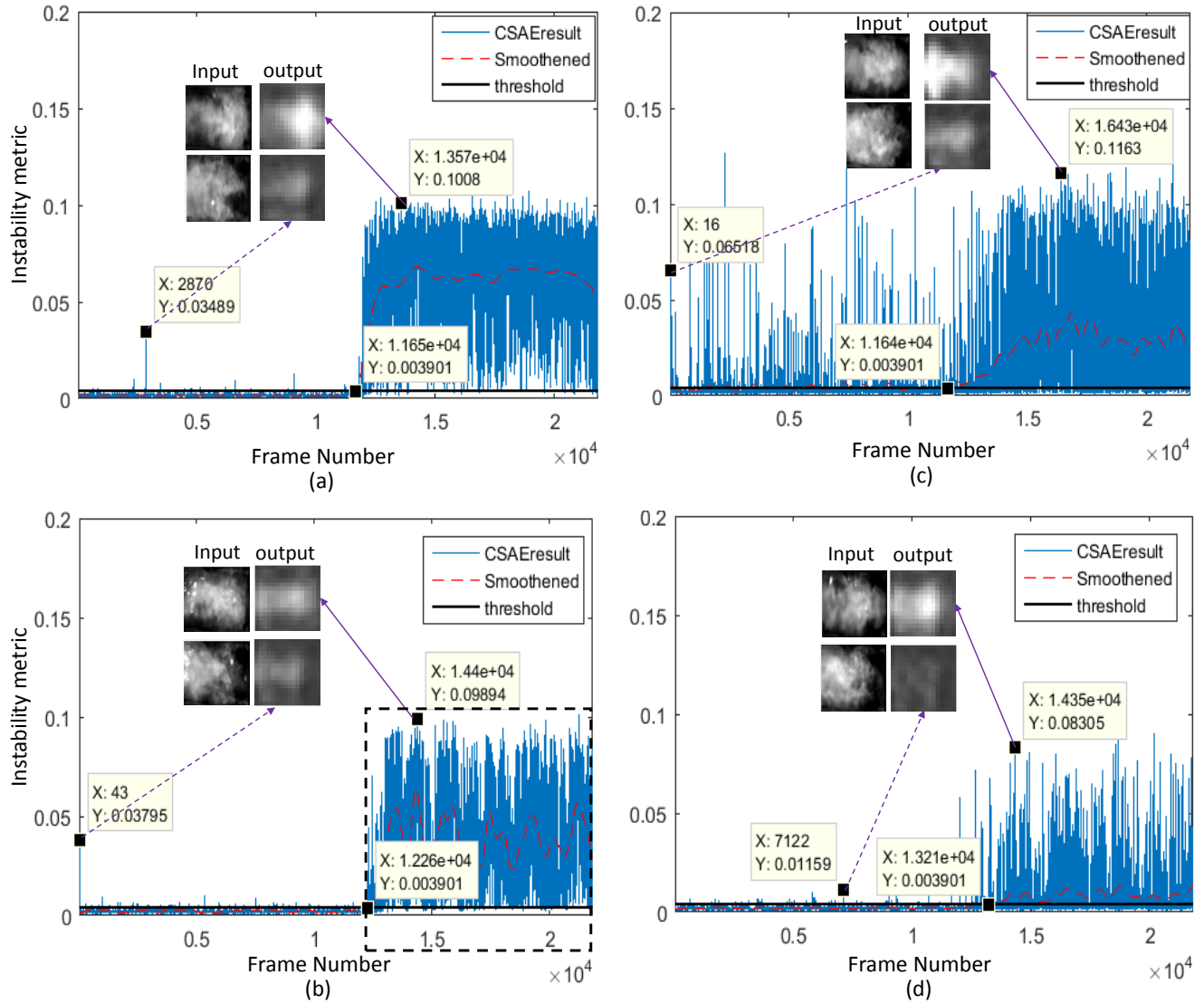
Figure 11. Results of transition protocols for: (a). 500_{40to30}, (b). 500_{40to28}, (c). $50_{700to800}$ and (d). $40_{500to600}$ where dashed arrows indicate the results for frames near the unstable flame in the transition region, and thick arrows show results for frames in the supposedly stable regions

Table 3. Performance metrics and transition start frame # for Transition Protocols

Protocol	CSAE				SCRF			
	$\mu(\mathbf{z})$	$\Sigma(\mathbf{z})$	$\frac{\Sigma(z)}{\mu(z)}$	frame #	$\mu(\mathbf{z})$	$\Sigma(\mathbf{z})$	$\frac{\Sigma(z)}{\mu(z)}$	frame #
500_{40to30}	0.0289	0.0302	1.045	11650	0.1449	0.0310	0.2139	–
600_{50to35}	0.0266	0.0378	1.4211	14470	0.1264	0.0380	0.3006	–
500_{40to28}	0.0175	0.0194	1.1086	12260	0.1417	0.0330	0.2329	–
$50_{700to800}$	0.0142	0.0125	0.8803	11640	0.1303	0.0150	0.1151	–
$40_{500to600}$	0.0045	0.0033	0.7333	13210	0.1262	0.0177	0.1403	–

5.4. Frame labeling

An extension of the algorithm's objectives could be made to implicit labeling. This is achieved by searching through all the frames to detect frames that are adjacent neighbors to a given frame. In clear terms, this means finding the label of a frame given the knowledge of the label of an adjacent flame. This kind of search is usually difficult with most primitive
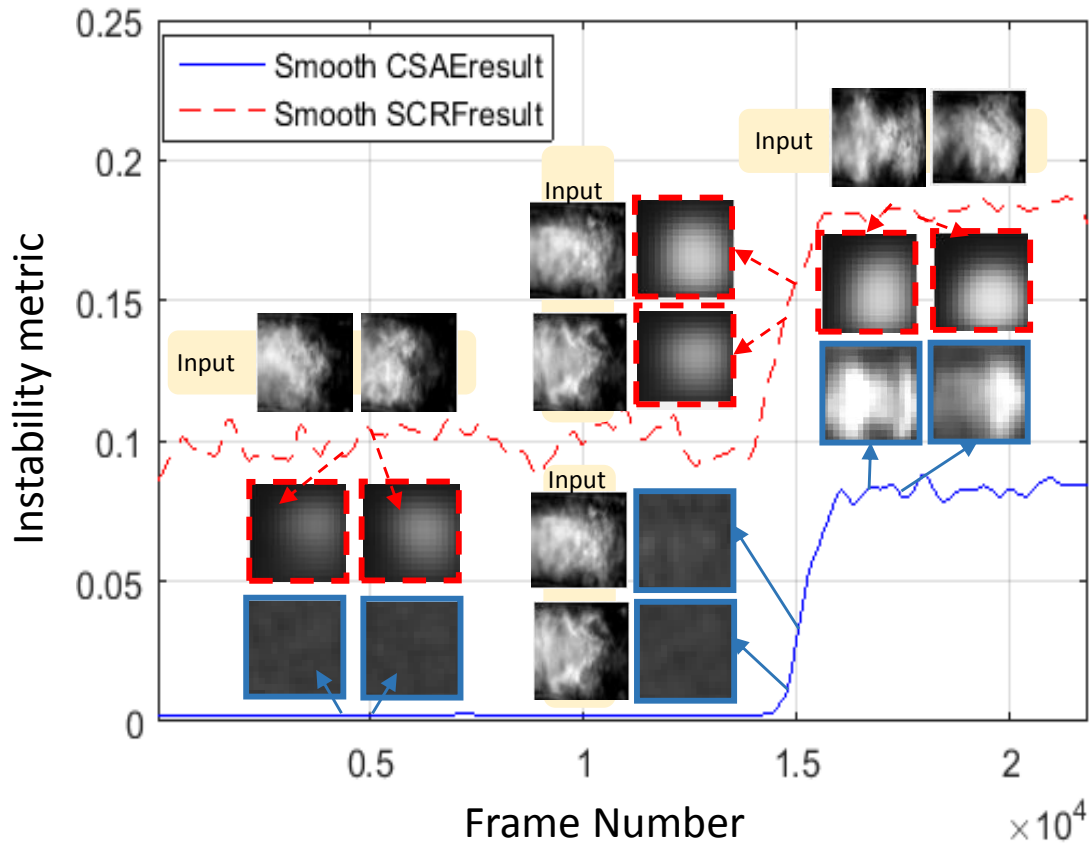
Figure 12. Adjacency labeling result of transition protocols for 600_{50to35} at the different regions of the profile. The image frames without any boundaries represent the inputs to the protocols at the points indicated by the arrows

low dimensional local labeling algorithms (e.g. HMM and MEMM) due to dependency depth and 'labeling bias' limitations respectively. For this purpose also, we were motivated to compare the results of CSAE to those of SCRF. We show that such high dimensional problem can be simplified with scalars that are calibrated in scales such as our instability metric. The highlighted examples in fig. 12 show how labeling may be achieved with the algorithms. Based on qualitative frame-to-frame visualization, labels provided by CSAE are shown in fig. 12 to outperform that of SCRF. CSAE is able to differentiate labels from frame to frame better than the CSAE in the separate flame regions. Frames in the region closest to UR have their the mushroom structures better labeled by CSAE while SCRF does not activate all the units for such labels. Importantly also, we find a gradual transition in the labels of frames in the almost linear transitioning stage of CSAE in much similar way as that of SCRF. Note that all input examples used for comparison in the figure are chosen at the similar frame numbers for both algorithms. The results provides briefly the potentials of the algorithm to deriving soft labels from intrinsically labeled classes, two classes in this case.

6. CONCLUSIONS AND FUTURE WORKS

An end-to-end convolutional selective autoencoder is developed to perform early detection of combustion instabilities using hi-speed flame video. Validation results are performed on data from a laboratory scale swirl-stabilized combustor. In addition to that, the framework was also used to generate fuzzy labels from prior knowledge of hard labeled examples as solution to implicit labeling problem. Conditional random field model results are used to compare the effectiveness of our deep learning based solution approach in both applications. Moreover, CSAE results shown confirm the expert's physical observation in the presence of coherent structures in stable flame regions. Some observed differences in the results are that: (i) CSAE is able to learn and generalize selectivity better than SCRF via more efficient masking the stable region; (ii) Unlike CSAE, SCRF introduces a bias in the instability metric computation for test data, such that its ability to act as an effective filter is hindered;(iii) SCRF succumbs to high false alarm rate during stable combustion. The fact that CSAE can detect instability early for various new (unseen in training phase) protocols while being trained on different pro-

tocols shows the generalizability of the proposed algorithm.

The results have been presented in the light of KL-distance based instability metric to determine the closeness to domain knowledge of stable flame frames reproduced by the models. Using the same metric, the architecture was extended to addressing the neighborhood implicit graph labeling problem. The framework can be generalized to soft-labeling of high-dimensional data. While the framework is shown to be an efficient diagnostics technique for combustion process in laboratory experiments, large scale validation is underway to demonstrate its wide-range applicability. Some of the future works are: (i) to extend the framework to labeling in multi-class scenarios; (ii) validation of possible coherent structures identified by CSAE in the transition region by using expert knowledge and fluid mechanics and (iii) to compute, in conjunction with other use-factors, the instantaneous estimate of the remaining useful life (RUL).

REFERENCES

Aharon, M., Elad, M., & Bruckstein, A. (2006). An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, *54*(11), 4311-4322.

Akintayo, A., Lee, N., Chawla, V., Mullaney, M., Marett, C., Singh, A., … Sarkar, S. (2016, March). An end-to-end convolutional selective autoencoder approach to soybean cyst nematode eggs detection. *arXiv*(arXiv:1603.07834v1), 1 - 10.

Akintayo, A., Lore, K. G., Sarkar, S., & Sarkar, S. (2016, March). Early detection of combustion instabilities using convolutional selective autoencoders on hi-speed video. *arXiv*(arXiv:1603.07839v1), 1- 10.

Barbu, A. (2009). Learning real-time mrf inference for image denoising. *IEEEXplore*, *978-1-4244-3991-1*(09), 1574-1581.

Berger, A. L., Pietra, S. A. D., & Pietra, V. J. D. (1996). A maximum entropy approach to natural language processing. *Association for Computational Linguistics*, *22*(1), 1 - 36.

Bergstra, J., Breulex, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., … Bengio, Y. (2010, June). Theano: a cpu and gpu math expression compiler. *Pro-*

ceedings of the Python for Scientific Computing Conference (SciPy). (Oral Presentation)

Berkooz, G., Holmes, P., & Lumley, J. L. (1993). The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, *25*(1), 539-575. doi: 10.1146/annurev.fl.25.010193.002543

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York, NY, USA: Springer.

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing:deep neural networks with multitask learning. *25th International Conference on Machine Learning*, 1 - 7.

Domke, J. (2013). Learning graphical model parameters with approximate marginal inference. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *35*(10), 2454 - 2467.

Duchi, J., Hazan, E., & Singer, Y. (2011, July). Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, *12*, 2121-2159.

Erdogan, H. (2010, December). *A tutorial on sequence labeling.* ICMLA.

Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2013). Learning hierarchical features for scene labeling. *exdb*, 1-15.

Fung, J., & Mann, S. (2004, August). Using multiple graphics cards as a general purpose parallel computer: Applications to compute vision. *International Conference on Pattern Recognition, ICPR*, *1*, 805-808.

Graves, A. (2014, June). Generating sequences with recurrent neural networks. *arXiv:1308.0850v5 [cs.NE]*, 1 - 43.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *IJCNN*, 1 - 8.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012, July). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv*(1207.0580v1), 1 - 18.

Hussain, A. K. M. F. (1983). Coherent structures - reality and myth. *Physics of Fluids*, *26*(10), 2816-2850.

Jones, S. (2015, April). *Convolutional autoencoders in python/theano/lasagne.* Retrieved Feb 03, 2016, from `https://swarbrickjones.wordpress.com/`

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *NIPS 2012: Neural Information Processing Systems, Lake Tahoe, Nevada*.

Kulesza, T., Amershi, S., Caruana, R., Fisher, D., & Charles, D. (2014, April). Strudtured labeling to facilitate concept evolution in machine learning. *ACM*, 1 - 10.

Kullback, S., & Liebler, R. (1951). On information and sufficieny. *The annals of mathematical statistics*, 79-86.

Lafferty, J., McCallum, A., & Pereira, F. C. (2001, June). Conditional random fields: Probabilistic models for

segmenting and labeling sequence data. *International Conference on Machine Learning*, 282-289.

LeCun, Y., & Bengio, Y. (1998). Convolutional networks for images, speech and time-series [Computer software manual]. MIT Press.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998, November). Gradient-based learning applied to document recognition. *Proc of IEEE*, 1-46.

Li, H., Zhou, X., Jeffries, J. B., & Hanson, R. K. (2007, February). Sensing and control of combustion instabilities in swirl-stabilized combustors using diode-laser absorption. *AIAA*, *45*(2), 1-9.

Liu, C., Ghosal, S., Jiang, Z., & Sarkar, S. (2016). An unsupervised spatiotemporal graphical modeling approach to anomaly detection in cps. *Proceedings of the International Conference on Cyber-physical Systems (IC-CPS)*, 1 - 10.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *CVPR*, 3431-3440.

Lore, K. G., Akintayo, A., & Sarkar, S. (2016, June). Llnet: A deep autoencoder approach to natural low-light image enhancement. *Elsevier Pattern Recognition*(doi:10.1016/j.patcog.2016.06.008), 1-13.

Masci, J., Meier, U., Ciresan, D., & Schmidhuber, J. (2011). Stacked convolutional auto-encoders for hierarchical feature extraction. In T. H. et al. (Ed.), (p. 52-59). Springer-Verlag Berlin Heidelberg.

Meyer, A. (2011-2012). *Hmm and part of speech tagging.* Lecture Note.

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech proccessing. *Proceedings of the IEEE*, *77*(2), 257-286.

Sarkar, S., Lore, K. G., & Sarkar, S. (2015, December). Early detection of combustion instability by neural-symbolic analysis on hi-speed video. In *Workshop on cognitive computation: Integrating neural and symbolic approaches (coco @ nips 2015).* Montreal, Canada.

Sarkar, S., Lore, K. G., Sarkar, S., Ramaman, V., Chakravarthy, S. R., Phoha, S., & Ray, A. (2015). Early detection of combustion instability from hi-speed flame images via deep learning and symbolic time series analysis. *Annual Conference of the Prognostics and Health Management Management Society*, 1-10.

Scherer, D., Muller, A., & Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. *Intenational Conference on Artificial Neural Networks, ICANN*, 1-10.

Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, *656*, 5-28. doi: 10.1017/S0022112010001217

Sercu, T., Puhrsch, C., Kingsbury, B., & LeCun, Y. (2016, September). Very deep multilingual convolutional neural networks for lvcsr. *ICAPPS*, 1–5.

Simonyan, K., & Zisserman, A. (2015, April). Very deep convolutionaal networks for large scale image recognition. *International Conference on Learning and Recognition*(arXiv:1409.1556v6), 14.

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. *International Conference on Machine Learning, JMLR*, *28*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going deeper with convolutions. *CVPR*, 9.

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. *Science*, *331*, 1279-1285.

Thoma, M. (2016, February). *Lasagne for python newbies.* Retrieved 03, from `https://martin thoma.com/lasagne for python newbies/`

Zeiler, M. D. (2012, December). Adadelta:an adaptive learning rate method. *arXiv:1212.5701v1*, 1-6.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. *ECCV*(8689), 813-833.

BIOGRAPHIES

Adedotun Akintayo was born in Lagos, Nigeria. He is currently a graduate student pursuing his Ph.D. in the Self-aware Complex Systems laboratory of Iowa State University. His research interest is in developing newer Machine Learning algorithms for extracting salient features to aid solving engineering and plant sciences problems. He earned his B.Eng. Mechanical Engineering degree at Ahmadu Bello University, Nigeria in 2010. Subsequently, he earned his M.Sc. degree in Control Systems at The University of Sheffield, United Kingdom in 2013.

Kin Gwn Lore was born in Selangor, Malaysia. He received his B. Sc. degree in Mechanical Engineering at Iowa State University in December 2014 and continued to pursue an M. Sc. degree in the same institution. His main research interests are machine learning methods (especially neural networks) in microscale device manufacturing, human-machine interaction systems, and image processing. He has co-authored a total of 8 peer-reviewed publications including 2 journal articles.

Soumalya Sarkar is a senior research scientist at United Technology Research Center (UTRC). He has received his B. Eng. Degree in Mechanical Engineering in 2010 from Jadavpur University, Kolkata, India. He received M.S. in Mechanical Engineering in 2013 and M.A. in Mathematics in 2014 from Penn State University. Dr. Sarkar received his Ph.D. in Mechanical Engineering from Penn State University in June, 2015. He has 5+ years of experience on Deep Learning, Anomaly Detection and Estimation, Sensor Fusion, Spatiotemporal Analysis, Situation Awareness, Statistical Signal Processing & Machine Learning and Probabilistic Graphical Models. He co-authored 27 peer-reviewed publications including 11 Journal papers, 2 book chapters and recently filed one US patent. Dr. Sarkar is a member of the American Society of Mechanical Engineers (ASME), Institute of Electrical and Electronics Engineers (IEEE) and PHM Society.

Soumik Sarkar is an Assistant Professor of Mechanical Engineering at Iowa State. He received his B. Eng. Degree in Mechanical Engineering in 2006 from Jadavpur University, Kolkata, India. He received M.S. in Mechanical Engineering and M.A. in Mathematics in 2009 and received his Ph.D. in Mechanical Engineering in 2011 from the Pennsylvania State University. Dr. Sarkar's research interests include Statistical Signal Processing, Machine Learning, Sensor Fusion, Fault Diagnostics & Prognostics, Distributed Control and Complexity Analysis with applications to complex Cyber-Physical Systems such as aerospace (e.g., gas turbine engines, traffic/communication networks, agricultural systems, manufacturing and energy systems (e.g., high-performance buildings, power grids). He co-authored 65 peer-reviewed publications including 23 journal papers, 3 book chapters and one magazine article. Dr. Sarkar served as a reviewer, organizer and session chair for several technical journals and conferences. He is currently serving as an Associate Editor of Frontiers in Robotics and AI: Sensor Fusion and Machine Perception journal. Dr. Sarkar is a member of the American Society of Mechanical Engineers (ASME) and the Institute of Electrical and Electronics Engineers (IEEE)