Application of computer vision and deep learning for flame monitoring and combustion anomaly detection

To cite this article: S Abdurakipov and E Butakov 2019 J. Phys.: Conf. Ser. 1421 012005

View the article online for updates and enhancements.
Application of computer vision and deep learning for flame monitoring and combustion anomaly detection

S Abdurakipov and E Butakov*
Kutateladze Institute of Thermophysics SB RAS, Prospekt Akademika Lavrent'eva 1, Novosibirsk, 630090, Russia
*E-mail: e_butakov@mail.ru

Abstract. This paper is devoted to study combustion of pulverized coal in a 5 MW thermal furnace with tangential scroll supply of coal-air suspension and cylindrical reaction chamber. Deep learning approaches were used to monitor the combustion of a coal flame in a furnace and to determine combustion anomalies from flame images. We have developed a deep neural network autoencoder, which is a combination of convolutional layers, fully-connected layers and upsampling layers. The autoencoder was trained to reconstruct the combustion regimes that corresponded to high values of the coefficient of excess air. The trained autoencoder was then used to identify abnormal burning regimes with a lower excess air ratio, in which there is an increase in the amount of unburned coal dust. The best classification model had AUC ROC quality metric value on a test image sample AUC ROC = 0.8. The average precision of the model was 77% and the average recall was 66%. The metrics obtained is limited by the quality of the image labeling due to poorly controlled experimental conditions. The paper concluded that the use of upsampling with convolutional layers show themselves better than the deconvolutional layers, and the combination of convolutional layers with fully-connected layers constitutes the optimal architecture of the model.

1. Introduction

Energy is one of the main life-supporting industries. At present, about 40% of the world's electricity is generated by coal-fired power plants [1]. The use of ordinary coal in power, especially in "small" energy, is associated with its inefficient combustion: low efficiency of existing boiler equipment and increased losses, unstable combustion, slagging of heating surfaces, higher harmful emissions into the atmosphere. Continuously growing energy consumption and environmental safety issues require maximum optimization of the process of coal combustion.

The development of modern machine learning algorithms has provided researchers and engineers with new effective tools for data mining of large complex multivariate databases and generalizing contained valuable information by building mathematical model of process. The application of such intellectual data driven approaches is relevant for improving the performance of technical devices, preventing failures and unplanned stops, because trouble-free operation of complex technical systems is one of the priorities of any production process [2-4]. Neural network autoencoders are widespread unsupervised machine learning algorithms whose primary goal is to find low-dimensional latent feature space. Autoencoders are widely used for a number of tasks, for example, noise reduction, image compression, dimension reduction, construction of a probability distribution model, etc. [5-7].
This work was aimed at studying the applicability of various neural network architectures of autoencoders to solve the problem of detecting anomaly combustion regimes of a coal flame in a combustion chamber. Abnormal burning regimes correspond to low values of the coefficient of excess air, in which there is an increase in the amount of unburned coal dust.

2. The experiment

The burning process were tested in a 5 MW pilot-scale combustion rig with dry (cyclone) slag removal (figure 1). Such a relatively large-scale set-up was chosen in order to study the possibility of using mechanically activated coal in industrial boiler and this experiment is the first step in the investigation of different conditions on the ignition and combustion stability. The rig consists of two stages. The first stage (denoted in the figure 1 (a) as burner) is a cylindrical swirl chamber of a diameter of 315 mm and length of 1515 mm into which micronized coal premixed with the primary air by an ejector was fed tangentially into the burner through a vaneless spiral. The second stage (denoted in figure 1 (a) as afterburner chamber or combustor) is a coaxial cylindrical duct with interior diameter of 1000 mm and length of 2820 mm, also equipped with a tangential entry for secondary air (which also reinforces the swirling motion). It is envisaged that additional ordinary pulverized coal can be fed together with the secondary air, but this option was not considered in the present experiment. The burner spiral has an aperture for a gas torch used to ignite the coal-air mixture. The main purpose of the first stage (burner) is the thermochemical preparation, ignition and sustenance of combustion of micronized coal so that it can ensure continued combustion in the furnace (chamber), but also the ignition and sustenance of additional non-activated pulverized coal that can be supplied to the afterburner chamber in real-scale installations. The main purpose of the secondary air was to ensure the complete burning of coal and its products (CO, CH) in the combustion chamber. Thus, its flow rate, while being monitored and adjusted to fulfill the above mentioned requirement, was not precisely controlled, as the main focus of the experiment was on the burner. The rig has adjustable systems for coal feeding, for primary and secondary air supplies, and for ash recovery. It is also equipped with a pressure controller in the combustion chamber.
Figure 1. Experimental rig of 5 MW power. (a) Schematics; T1–T8 denote thermocouples and G5 – G7 the gas sampling apertures. (b) View at the furnace (burner and combustion chamber) in autothermal operation. Inspection windows show stable and uniform flame along the whole combustor length.

For combustion monitoring, the rig was also equipped with a number of apertures along the burner and combustion chamber for thermocouples and gas sampling probes. The temperature was measured by platinum-rhodium thermocouples of type B (TPR/1-0679, ‘Vakuummash’, Izhevsk, Russia), with 30% and 6% of rhodium in one and the other conductor respectively. The B type thermocouples have proven to have superior thermoelectric and tensile properties, and to be exceptionally stable and resistant to oxidizing atmosphere at high temperatures. The wires of 0.3 mm thickness are placed in a ceramic sheath of 4 mm in diameter. The tip of the ceramic sheath is recessed with a 0.5×3 mm slot so that the thermocouple junction is directly exposed to the measuring medium. (The external ceramic sheath of 10 mm diameter, provided by the manufacturer for measurements in molten metals, was removed). The calibration data, provided by the manufacturer and approved by the Russian National Standards (GOST), complying with the international standard (IPTS-68) guarantee the nominal temperature range of 600–1650 °C for long continuous operation (up to 6000 h) with accuracy of ±0.004t, where this the measured temperature in °C. The measuring range can be extended to 1800 °C for a shorter use. It is noted that none of the thermocouples was used longer than 10 h in total, with a typical measurements campaign lasting no longer than 15 min.

The gas samples probes are connected to a multi-component (O2, CO2, CO, CH NO, NO2) gas analyzer (Boner-VT) with real-time computer processing of the measured data.

To study the burning process, 5 MW thermal furnace with tangential scroll supply of coal-air suspension and cylindrical reaction chamber was used. Brown coal (one of the main coal deposits of Siberia) was used in experiments. The technical composition of coal is as follows (mass %, dry):
moisture $W_d = 30.8$, ash $A_r = 11.1$, volatile content – 49.1, sulfur $S_d = 0.29$, high heat value $Q_{daf} = 6900$ kcal/kg. Coal was fed to the high-energy mills and then was directed by an ejector with transport air to the scroll burner.

3. Introduction to algorithms

Neural autoencoder – unsupervised machine learning algorithm. The main idea of the autoencoder is to reduce the dimension by mapping the original feature space into the latent feature space of a smaller dimension (encoding step), and then in reconstructing the input data based on the latent representation (decoding step) (see figure 2). By reducing the dimension of feature space, we thereby train the model to learn only the most important information from which initial data can be reconstructed [3].

Autoencoders consist of two parts: encoder $g$ and decoder $f$. The encoder transforms the input signal into its representation (code): $h = g(x)$, and the decoder reconstructs the signal by its code: $x = f(h)$. The autoencoder, varying $f$ and $g$, tends to learn the identity function $x = f(g(x))$, minimizing some error functional $L(x, f(g(x)))$. Property of dimension reduction is achieved by forming the encoder structure in the form of a bottleneck, significantly reducing the dimension of the last hidden layer. The goal of the autoencoder is to minimize the reconstruction error between the input and output data. To reduce the reconstruction error, during training process we “propagate” in the opposite direction along the neural network and update the weights of the neural network. Weights of neurons are updated depending on how much they contribute to the reconstruction error. Autoencoder can be considered as a data compression algorithm [5]. At the same time, the families of functions of the encoder $g$ and decoder $f$ are somehow limited, so that the autoencoder is forced to select the most important properties of the signal. The critical limitation of autoencoders is that the data should belong to the general population on which the network was trained. Having trained the autoencoder on digit numbers, it cannot be used to encode something else (for example, human faces). Moreover, some of their developments, such as the variational autoencoder (VAE), as well as its combination with generative adversarial networks (GAN), give very interesting results and are now at state-of-the-art of generative models.

![Figure 2. Scheme of autoencoder [5].](image)

The neural autoencoder captures the non-linear features of the data, and therefore has an advantage, for example, over the linear principal component analysis approach (PCA). To build an autoencoder, it is necessary: 1) to define a coding method, 2) a decoding method, 3) a loss function for comparing the output of the autoencoder with the target result. In this paper, the root-mean-square error was used as a loss function. The simplest version of a neural network autoencoder is an encoder-decoder structure consisting of fully-connected layers, without feedback, most similar to a perceptron. A more complex option, more suitable when working with images is to replace full-connected layers with convolutional layers. A convolutional layer is a set of maps; each map has a convolution operator or filter. The filter
moves through the image, multiplies the filter values by the original pixel values of the image (elementwise multiplication). All these multiplications are summed, and as a result one number is obtained—the result of convolution. The number of feature maps is determined by the requirements of the task; if you take a large number of feature maps, the recognition quality will increase, but the computational complexity will increase too. Based on the analysis of scientific articles, in most cases it is proposed to take the ratio of one to two, that is, each feature map of the previous layer (for example, the first convolutional layer, the previous one is the input one) is associated with two maps of the convolutional layer.

One of the important implementations of this paper is the use of upsampling layers in autoencoder architecture. Inverting a convolutional layer can be quite complicated due to overlapping of the kernels. Most deep learning architectures use deconvolutional layers (sometimes referred to as transposed convolution layers), which are designed to invert the convolutional layer. Even though the transposed convolution layers are intuitive for the reconstruction of the input data, they have a few well-known drawbacks, such as the generating checkboard artifacts. To overcome this limitation, the transposed convolution layers were replaced with oversampling layers and simple convolutional layers. We also tested this solution on a handwritten number dataset.

4. Model description

During the study, we worked on the following neural network architectures:

1) Simple autoencoder with five hidden fully-connected layers (basic model);
2) Convolution autoencoder, which consists only of convolutional layers in the encoder and transposed convolutional layers (or deconvolutional layers) in the decoder;
3) Another convolutional model that uses convolution and subsampling layers in the encoder, and upsampling with convolutional layers in the decoder;
4) Combination of convolutional and fully-connected layers.

To simplify the implementation of the neural network, we used Python development using Keras framework with Tensorflow backend. A simple fully-connected autoencoder had the architecture: Input (128 × 128) – Flatten (16384) – Dense (1024) – Dense (1024) – Dense (512) – Dense (1024) – Dense (16834) – Reshape (128 × 128), where Input is the layer of input parameters, Flatten is a straightening operator, Dense is a fully-connected layer, Reshape is a shape changing operator. The second model is a convolutional autoencoder, which consists only of convolutional and deconvolutional layers. In the encoder, the input data passes through 12 convolutional layers (Conv2D) with 3 × 3 cores and filter sizes starting from 8 and increasing to 32. The decoder reconstructs image using transposed convolutional layers (Conv2DTranspose). Using only convolutional layers may seem unjustified, but in this case the goal is to compare methods instead of achieving the highest possible accuracy. The latter architecture, which was analyzed, is a convolutional autoencoder with convolutional layers, pooling layers, and upsampling layers (see figure 2.). The encoder is the following sequence of layers: Conv2D – Con2D – MaxPooling2D – Conv2D – Conv2D – MaxPooling2D – Flatten – Dense. Thus, alternating convolutional (Conv2D) and pooling layers (MaxPolling2D), ending in a fully-connected layer (Dense). At the exit from the last convolutional layer of the decoder, a reconstructed image with an initial resolution of 128 × 128 is obtained.

Data preparation. Most of the original image containing the shaded area has been cropped. The original data resolution has been reduced to 128 × 128 pixels. The data were split into a training, test and validation sample. In the training sample got 2000 images, in the test 200 images and 200 in the validation. In the training sample, there were only combustion regimes with a large excess fuel ratio, which are characterized by a high burning intensity of the coal torch.
Figure 3. The architecture of the combined model of autoencoder (a) encoder and (b) decoder.

Ideally, we want to train the neural network so that it remains resistant to image distortions and noise in the data, but our model can be trained only on the basis of those samples that we provided to it, despite the fact that it performs some kind of statistical analysis of the training set and extrapolates him. In this paper, we applied a simple but effective solution: to artificially expand the training data.
with distorted versions of images during training. This means the following: before setting an example for the model input, we apply all the transformations we deem necessary, and then let the network directly observe what effect it has on applying to the data and training it to behave well on these examples. For example, here are some examples of shifted, scaled, deformed, rotated images (see figure 4). To implement this procedure, we used Keras framework, which provides a valuable interface for expanding the training set – ImageDataGenerator class. We initialize data generator and choose, what types of transformation we want to apply to the images, and then we run the training data through the generator, calling the fit method, and then the flow method, getting a continuously expanding iterator for the batch types we replenish. There is even a special model.fit for the batch types we replenish. There is even a special method model.fit_generator that will teach our model using this iterator, which greatly simplifies the code. ImageDataGenerator also provides us with the ability to perform horizontal vertical shifts, random turns, scaling, warping, and specular reflection.

![Figure 4. Examples of augmented images in training set. Shifts, small angle rotations, blur, zoom were applied.](image)

*Training.* Neural network optimization method, hyperparameters, batch size, number of learning epochs, etc. We used the ModelCheckpoint procedures to monitor and preserve the neural network in training process and EarlyStopping procedure for loss function on a validation sample with a patience stop = 10 (training was stopped if the loss function did not decrease within 10 epoch). Number of epochs = 500, batch_size = 8. The mean square error (MSE) was used as a loss function. Adam's adaptive inertia method was used as an optimization method, which is currently the standard option for most tasks.

*Metrics.* After calculating the root-mean-square reconstruction error, it was normalized to one for its interpretation as the probability of belonging to the class of anomalous images. The higher the probability, the higher the confidence of the classification model. Then, the cut-off was selected according to the probability at which the smallest number of false positives of the model was observed.
and the quality metrics were calculated. Precision, recall and areas under ROC and PRC curves were used as quality metrics. The ROC curve is a very useful metric for assessing the quality of binary classifiers, and the area under it is the most popular feature of the quality of binary classifiers [7]. AUC ROC is equal to proportion of image pairs (image of class 1 – anomaly, image of class 0 – normal), which the algorithm correctly ordered, i.e. the first object goes in the ordered list before. The ROC curve reflects the dependence of true positive rate versus false positive rate. The advantage of ROC curve is that the area under the curve does not depend on the probability cutoff. We want the blue curve to be as close as possible to the upper left corner, then the area under the curve will tend to unity. However, due to the imbalance of classes, ROC curve may not be indicative. Therefore, we also use precision and recall metrics, as well as the area under of precision-recall curve (PRC). Precision is the proportion of images that belong to this class relative to all images that the model has assigned to this class. The recall is the proportion of images found by the classifier that belong to the class relative to all images of this class in the test set. The area of the precision-recall curve may be more indicative than AUC ROC in the case of strong class imbalance, but this quality metric is not interpretable, unlike AUC ROC.

5. Results

Our model seems to catch a lot of abnormal burning situations, when a large amount of coal dust is present when burning. However, the number of ordinary situations classified as an anomaly is indeed large, but the sensitivity of the detector can be adjusted by varying the probability threshold. The average classification precision was 77% and average recall was 66%. The area under precision-recall curve was AUC PRC = 0.54.

Figure 5. Reconstruction error on test sample. The blue dots show the reconstruction error of normal images and red ones – anomaly images. The classification probability is proportional to the reconstruction error.

Figure 5 shows the error of reconstruction of images of an autoencoder trained by using only normal images. As a result, it can be seen that the error in reconstructing images with high values of the coefficient of excess air is much lower compared with the regimes at low values of the coefficient. Thus, based on the analysis of the reconstruction error, it is possible to classify the normal and abnormal regimes of coal combustion. If the reconstruction error is scaled to unit interval, then it can
be interpreted as the probability or degree of confidence of the classifier when the image is assigned to one or another class. The cutoff for the root-mean-square error equal to 200 is selected. Figure 6 shows the ROC curve and classification error matrix. The area under the ROC curve, demonstrating the quality of the binary classification is 0.8. According to the analysis of the model’s confusion matrix, it can be seen that the model has a sufficiently large number of false negatives, that is, we do not detect a significant part of the abnormal images. Figure 7 shows the output of a fully-connected layer of the encoder, demonstrating how the model separates normal and anomalous observations. Basically, the model relies on the size of the region of intense luminescence on the periphery. The resulting number of errors is probably due to errors when labeling images due to insufficient control of parameters during the experiment. In Figure 7 image (a) model correctly classifies to normal. The image (c) is correctly assigned to the abnormal state, which is probably associated with a small area of intense luminescence from combustion, which is especially noticeable at the periphery. This seems logical because the intensity and size of the burning area is related to the amount of unburned coal dust. Image (b) is anomalous, but the model falsely classifies it as a normal combustion regime.

Tuning. Usually the process of developing a neural network begins with the development of a simple network (baseline model), either directly applying those architectures that have already been successfully used in literature to solve similar problems or using those hyperparameters that have previously produced good results. Ultimately, we hope, we will reach a level of performance that will serve as a good starting point, after which we can try to change all the fixed parameters and extract the maximum performance from the network. This process is commonly referred to as setting up hyperparameters, because it involves changing the network components that must be determined before starting the training. To improve the quality of models, the following approaches will be applied in the next step:

1) Regularization. One of the main problems of machine learning is overfitting, when the model, in pursuit of minimizing the cost of training, loses the ability to generalize. Regularization makes it possible to reduce the complexity of the model. One of the effective ways to regularize a neural network is Dropout approach. A more direct approach to regularization, this is a penalty for a higher weights rate (L2 regularization) [8].

2) Weight initialization. When calculating initial weights of neural network, one can rely on a probability distribution (uniform or normal) with a dispersion equal to \( D = \frac{2}{(n_{in} + n_{out})} \), where \( n_{in} \) and \( n_{out} \) are the number of neurons in the previous and next layer, respectively. The

![Figure 6. ROC curve and confusion matrix.](image-url)
He initialization method is more suitable for the ReLu activation function, compensating for the fact that this function returns zero for half the definition domain \(D = 2 / n_{in}\) [9].

**Figure 7.** Visualization of output of fully-connected layer for neural network autoencoder. Examples of flame images in the combustion chamber as the excess fuel ratio decreases (from left to right). Image (a) is a true classified normal observation. Image (c) is a true classified anomaly. Image (b) is a false negative error.

3) **Batch normalization.** One of the popular methods for accelerating deep learning. The method solves the following problem, which impedes the effective training of neural networks: as the signal propagates through the network, even if we normalize it at the input, passing through the inner layers, it can be greatly distorted both by expectation and dispersion, which is fraught with serious inconsistencies between gradients at various levels. Therefore, we must use stronger regularizers, thereby slowing down the pace of learning. Batch normalization offers a very simple solution to this problem: to normalize the input data in such a way as to get a zero expectation and a unit variance. Normalization is performed before entering each layer.

4) **Cyclic learning rate.** The learning rate is one of the most important hyperparameters during training the neural network. A recent idea is called the cyclical learning rate, which is to increase the learning rate from time to time. This procedure allows to find another local minimum if the current minimum is not sustainable, and makes the model better generalized with new data [10].

6. Conclusions

In this work, a study was conducted of the possibility of using computer vision approaches, in particular, deep neural networks to detect abnormal burning regimes of a coal dust in the combustion chamber, corresponding to low values of the air excess coefficient, according to the analyzed gas composition. We have developed a deep neural network autoencoder, which is a combination of convolutional layers, fully-connected layers and upsampling layers. The autoencoder was trained to reconstruct the combustion regimes that corresponded to high values of the coefficient of excess air. The trained autoencoder was then used to identify abnormal burning regimes with a lower excess air ratio probably associated with an increase in the amount of unburned coal dust. The best classification model had AUC ROC quality metric value on a test image sample AUC ROC = 0.8. The average precision of the model was 77\% and the average recall was 66\%. Despite the large number of false negatives, the model can be useful for detecting anomalies in the combustion chamber. The precision and recall of the model can be significantly enhanced by collecting more detailed data with controlled combustion parameters, which will make the image labeling more accurate. The paper concluded that the use of upsampling with convolutional layers show themselves better than the deconvolutional layers, and the combination of convolutional layers with fully-connected layers constitutes the optimal
architecture of the model. In addition, complementary regulariztion methods could help generalization, but this seems unnecessary, since a convolutional autoencoder with layers of high sampling could achieve almost as good results with a network ten times smaller. The combination of these two types of layers in the end ensured the best performance with a reasonable architecture.

Acknowledgements
This work was carried out within the framework of the state assignment for the IT SB RAS.

References
[1] Chernetskiy M, Burdukov A, Butakov E, Anufriev I and Strizhak P 2016 Using ignition of coal dust produced by different types of mechanical treatment under conditions of rapid heating Combustion, Explosion, and Shock Waves vol 52 pp 326–28
[2] Abdurakipov S, Gobyzov O, Tokarev M and Dulin V 2018 Combustion Regime Monitoring by Flame Imaging and Machine Learning Optoelectronics, Instrumentation and Data Processing vol 54 pp 513–19
[3] Abdurakipov S and Butakov E 2018 Combustion anomalies detection for a thermal furnace based on Recurrent Neural Networks Journal of Physics: Conference Series vol. 1105 012043
[4] Gobyzov O, Tokarev M, Abdurakipov S and Lobasov A 2018 Flame state diagnostics using visualization and neural network analysis AIP Conference Proceedings vol 2027 040067
[5] Hinton G and Salakhutdinov R 2006 Reducing the dimensionality of data with neural networks Science vol 313 pp 504–7
[6] Bengio Y, Courville A and Vincent P, 2013 Representation learning: A review and new perspectives IEEE Transactions on Pattern Analysis and Machine Intelligence vol 35 pp 1798–828
[7] Wen L, Gao L and Li X 2017 A new deep transfer learning based on sparse auto-encoder for fault diagnosis IEEE Transactions on Systems, Man, and Cybernetics: Systems vol 99 pp 1–9
[8] Xu J, Xiang L, Liu Q, Gilmore H, Wu J, Tang J and Madabhushi A 2016 Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images IEEE transactions on medical imaging vol 35 pp 119-30
[9] Goodfellow I, Courville A and Bengio Y 2016 Deep learning The MIT Press Cambridge
[10] Smith L 2017 Cyclical learning rates for training neural networks IEEE Winter Conference on Applications of Computer Vision pp. 464–72