CNN-based Task State Estimation for Safer Automation of Oxy-fuel Metal Cutting

James Akl[1], Shreedhar Kodate[2], and Berk Calli[1]

***Abstract*— The industrial operation of oxy-fuel metal cutting via gas torches involves tasks such as ignition, preheating, and combustion along the target surface. Automated oxy-fuel cutting systems are exposed to risks and anomalies that can lead to incorrect actions and safety hazards. In this paper, we develop a classifier for online task state estimation to assess the cutting robot's actions, detect anomalies, and reduce the risk of hazards. Using representative footage from our robotic cutting experiments, we curate an image dataset labeled with four types of cutting task states. Using deep learning methods, we design and train a convolutional neural network model for classifying the cutting task state from input images. The classifier architecture is optimized for rapid inferences during online estimation. After evaluation, our classifier achieves an overall accuracy of 93.8% with high inference speeds on two types of representative hardware. Our 'Oxy-fuel Cutting Task State' (OCTS) dataset is available at `doi.org/10.5281/zenodo.7734951`.**

I. Introduction

The global economy of the future is projected to become increasingly automated. The industrial sector is particularly subject to pro-automation market pressures such as falling prices of automation systems [1]. However, scaling up the automated economy comes with its unique challenges. In particular, the rising importance of workplace safety [2] imposes stricter safety requirements in the design and adoption of automated systems [3]. This is emphasized in the automation of hazardous work that is highly-exposed to risk, such as metal cutting and welding. In parallel, there is considerable incentive to automate the difficult processes of oxy-fuel cutting [4] and welding. In existing work, the focus is often to achieve some autonomous functionality while safety is not sufficiently addressed.

In this work, we develop a classifier for task state estimation to monitor automated oxy-fuel cutting systems and detect anomalies for the benefit of increasing workplace safety. For this, we curate an image dataset of oxy-fuel metal cutting scenarios, representative of key task states in oxy-fuel cutting. The images are obtained from a series of recorded cutting experiments performed with a torch-equipped robot that cuts steel plates. These images are then labelled into distinct task state classes, identified by their dominant visual feature ('Torch flame', 'Preheating pool', 'Combustion pool', and 'Not applicable'). Using this data, we adopt a deep learning approach for capturing the data's feature hierarchy via a convolution neural network (CNN) model.

The purpose of this classifier is to monitor live oxy-fuel cutting operations. The classifier receives online vision data and infers the cutting task state, providing an interpretation to the robot (Fig. 1). When the classifier's interpretation and the robot's internal task state agree, an anomaly is unlikely and the robot proceeds with its actions. Conversely, a discrepancy between them suggests the occurrence of an anomaly in which case cutting operations are halted for safety inspection and response. The aim is to help validate the robot's actions for the detected task state. By monitoring the task environment and signaling discrepancies with the robot's actions, the classifier adds one layer of safety.

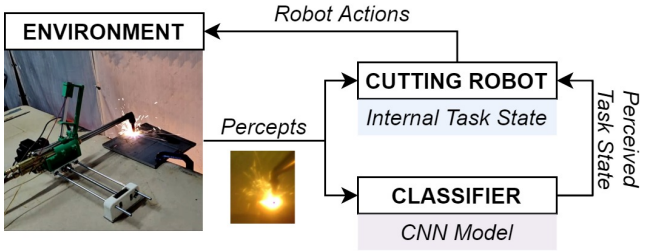

Fig. 1. The classifier interprets the environment percepts and provides the perceived task state to the cutting robot. When the internal and perceived task states match, the robot actions are validated and executed. Upon disagreement, an anomaly is declared and cutting operations are halted.

The core contributions of this work are:

1) Creating an image dataset for oxy-fuel cutting task states, obtained from automated cutting experiments.
2) Developing a CNN model to classify the task state from input images with high inference speed.
3) Evaluating the classifier's performance against a separate test set previously unseen by the classifier and identifying its strengths and limitations.

To the best of our knowledge, this is the first study in the literature focusing on vision-based state estimation of oxy-fuel cutting operations. Furthermore, this work publishes the first dataset in the literature covering the task states of oxy-fuel cutting and the first treatment in developing an appropriate classifier for these task states.

II. Related Work

Industrial processes using welding, cutting, and laser tooling are studied using a variety of instrumentation and techniques to extract higher-level perceptual information from lower-level sensory data. Often, the processed area (which may exhibit a heat pool or melt pool) is monitored and characterized to model or predict its effect on process quality, anomaly and defect detection, or penetration depth. Non-learning based techniques can be used to characterize combustion, *e.g.*, for the efficient operation of industrial

[1]Robotics Engineering Department and [2]Data Science Program, Worcester Polytechnic Institute, 27 Boynton Street, Worcester, MA 01609, USA.
E-mail: {jgakl, sskodate, bcalli}@wpi.edu

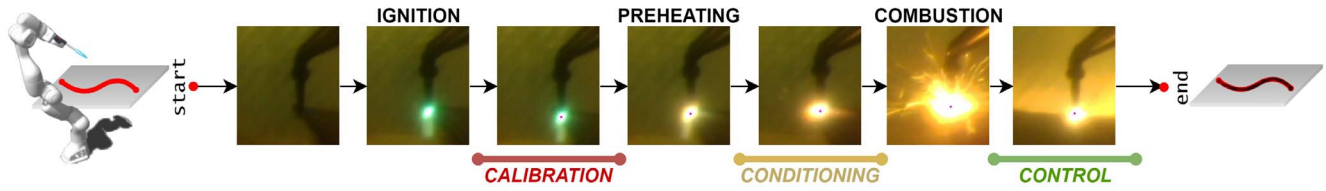

Fig. 2. Sequence of events (labeled in black) and tasks (labeled in color) in vision-based autonomous oxy-fuel cutting. Conceptually, the robot is equipped with a vision sensor and cutting torch and must cut the target object along the desired cutting path. The cutting events are the instances that trigger the cutting tasks. The footage shown is retrieved from vision-based cutting experiments using our 1-DOF cutting robot.

furnaces using classical image processing [5] and 3D instrumentation [6]. However, these approaches are often limited to processes that can be explicitly modeled.

More recent advances focus on applying learning-based approaches to infer task-relevant information while generalizing to a broader range of input scenarios. Indeed, the aim of learning-based applications to is to enable or improve industrial processes by monitoring for errors and ensuring that the goal states are reached. Such data-driven and learning-based methods are successfully applied to a wide variety of media: gas tungsten arc welding (GTAW) [7]–[11], gas metal arc welding (GMAW) [12], [13], submerged arc welding [14], variable polarity plasma arc welding [15], laser welding [16]–[18], wire-arc additive manufacturing (WAAM) [19], [20], wirefeed laser additive manufacturing [21], and orthogonal metal cutting [22].

In effect, neural network models are used abundantly for such industrial processes. While diverse neural architectures are adopted, convolutional neural networks [9], [10], [13], [16], [18], [20], [21], [23], [24] are most encountered in these applications. Often, pre-trained models are used or adapted such as the ResNet architecture for its deep convolutional layers and residual connections [17], [19]. Other approaches applied include generative adversarial networks [12], ensemble methods [11], autoencoders [22], vision transformers [8], transfer learning [7], and extreme learning machines [15].

In practice, while the problems encountered can exhibit similarities, they also carry distinct challenges due to the particularities of their tooling and the differences in their data. These domain-specific distinctions not only suggest preference for certain learning models, but also inform about auxiliary techniques that exploit domain knowledge. For instance, X-ray imaging can reveal more information for certain weld defects [18], [23]. Similarly, problem-specific advantages arise with other auxiliary techniques such as multimodal sensing [12], [16], [19], multisource sensing [11], image preprocessing [7]–[10], and acoustic sensing [13].

The following problems in prior work are most related to ours. For GTAW and weld pool image data: penetration classification (3 classes) in [11] and prediction in [10]; pool classification (2 classes, 6 classes) for defect identification in [7]; and, penetration state classification (4 classes) for penetration recognition in [8]. For GMAW and weld pool image data: pool state classification (4 classes) for defect detection in [12]. For WAAM and melt pool image data: pool state classification (4 classes) for anomaly detection in [20].

While there are similarities, our problem is distinct for the following reasons. (1) The industrial process is oxy-fuel metal cutting producing a heat pool from combustion, not a weld pool or melt pool. (2) The image data is particular to the oxy-fuel cutting medium and its associated events and tasks. (3) The problem is to classify the cutting task state for monitoring the robot actions' safety and correctness. (4) The classifier's inference time must be sufficiently low for online estimation, thus constraining its design and architecture.

Moreover, the aim of task state estimation is to enable higher-level reasoning about the robot's actions from lower-level data. For instance, [25] applies this strategy to robot contact tasks, extracting high-level action grammars from low-level trajectory data. To the best of our knowledge, this is the first work applying deep learning to automated oxy-fuel cutting for classifying its task states.

III. Problem Formulation

This section delineates the problem elements: the process and tooling of oxy-fuel cutting, the events and tasks relevant to its automation, and the role of task state estimation.

A. Oxy-fuel Cutting

The operation of oxy-fuel cutting consists of manipulating a cutting torch along a metal surface to cut through it along a desired path. The torch flame is produced by burning an oxy-fuel gas mixture, where the fuel is typically acetylene or propane. Material removal is usually achieved via a combustion reaction with the metal (most commonly carbon steel). This requires the metal surface to be sufficiently preheated using the torch flame. During preheating, the heated region on the metal surface exhibits the formation of a heat pool. This apparent bright blob is an accumulation of heat where combustion is most intense. Upon sufficient preheating, combustion is intensified by increasing oxygen flow and the torch is moved along the desired cutting path at an adequate velocity to maintain adequate conditions for combustion and material removal.

B. Automated Cutting

The oxy-fuel cutting process is complex and involves several events and tasks. Its automation may be tackled via different approaches, depending on the sensing modality and the desired degree of autonomy. We focus on vision-based automated cutting due to the visual stimuli produced

by the heat pool and the vision-based tracking inspired by the techniques of skilled cutting workers. This requires a representation of the cutting problem tailored to the vision-based approach of its automation.

The sequence of events and tasks during vision-based automated oxy-fuel cutting is illustrated in Fig. 2. The key events during the cutting process are: (1) *Ignition*: The torch flame is ignited and focused; (2) *Preheating*: The flame is positioned at the surface; and, (3) *Combustion*: Oxygen flow is increased via a lever. Accordingly, the cutting system executes these tasks: (1) *Calibration*: Calibrate the vision system against the torch's flame; (2) *Conditioning*: Heat the surface to combustion conditions; and, (3) *Control*: Regulate the torch motion for combustion cutting.

C. Task State Estimation

The above sequence of tasks for vision-based cutting, while effective, executes under the assumption of ideal operational conditions. When the expected cutting event is detected, the task state is updated and the robot continues onto its next action. In practice, oxy-fuel cutting operations are exposed to various potentials errors, risks, and hazards.

We enumerate instances of potential failure modes: ignition may fail; the flame may not be correctly focused; calibration may fail due to excess noise; combustion may fail fault of insufficient conditioning; faulty oxy-fuel tooling and leaks; excess sparks, slag, fire, or other anomalies; electronic or mechanical failures, and software errors; among others.

Such failures would manifest as anomalies in the image frame and the heat pool. While the automated system possesses internal state of its actions, it lacks external state of its environment. As such, under anomalous conditions the cutting agent's actions may be incompatible with the state of its environment. Under such cases, there may occur inefficiencies, failures, hazards, and dangers without intervention.

To address this, the cutting system tasks must be monitored and validated. This can be achieved using a classifier that infers from the robot's visual input one of four task states identified by their dominant feature in the image: 'Torch flame', 'Preheating pool', 'Combustion pool', and 'Not applicable'. This classifier is trained on representative footage, labeled with the desired task states. During cutting, the robot would check its internal task state with the classifier before taking action (see Fig. 1). Using this task state estimation, the robot gains some external awareness and performs safer cutting operations.

IV. Oxy-fuel Cutting Dataset

To train and test the task state classifier, we develop a dedicated dataset that captures the particularities of automated oxy-fuel cutting. This need is reinforced due to the lack of relevant datasets that are publicly available. Our 'Oxy-fuel Cutting Task State' (OCTS) dataset can be accessed at **doi.org/10.5281/zenodo.7734951**.

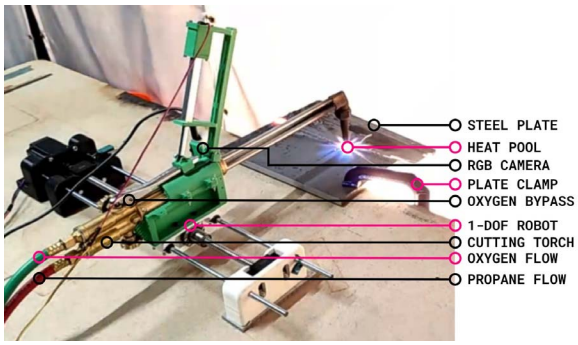

Fig. 3. The 1-DOF cutting robot performs cutting experiments wherein its RGB camera records the affected area on the steel plates. We note that the camera is mounted at a fixed pose towards the torch tip and that its lens is covered with a tinted visor (as is worn by skilled cutters). This tinted visor dims the scene, focuses on the flame and the pool, and prevents image saturation due to the extreme brightness of the flame and the pool.

TABLE I
Summary of the Data Collection Parameters

Total Experiments	50	**Frames Collected**	142671
Footage Recorded (min)	119	**Frame Dimensions** (px)	(640, 480)
Recording Rate (fps)	20	**Frame Channels**	(R, G, B)

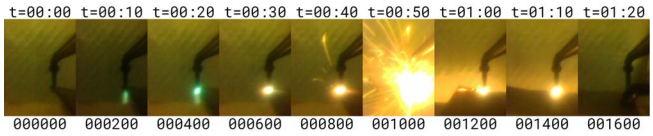

Fig. 4. Sequential footage from a particular experiment is shown at intervals of 200 frames (or 10 second). This particular cut is executed without fault.

TABLE II
Summary of Data Labels and their Associated Task States

Label	Element	Nominal Task State
TF	Torch flame	Calibrate the vision system.
PP	Preheating pool	Condition the metal surface.
CP	Combustion pool	Control the torch motion.
NA	Not applicable	Halt the cutting operations.

A. Data Collection from Cutting Experiments

Using our automated setup shown in Fig. 3, we perform cutting operations (as in Fig. 2) on steel plates. The area of interest on plates is recorded using the robot's eye-in-hand RGB camera. In total, 50 experiments across 11 sessions are recorded using the Intel RealSense D435 yielding 142,671 collected images. Some cuts are completed without fault (Fig. 4), while others are subjected to diverse failure modes. Depending on the severity of the failure, cuts are either fully completed, or partially completed due to interruptions. In this manner, the footage spans a broad range of representative, varied, and diverse cutting conditions. The data collection parameters are summarized in Table I.

B. Data Labels & Classification Problem

The classification problem is based on identifying the prominent element in the image: the torch flame (TF), the preheating pool (PP), or the combustion pool (CP). In addition, the label NA corresponds to 'Not applicable'

Fig. 5. Sample frames for each class from various experiments. We note the importance of covering variations of the key elements and ambient conditions.

where no dominant element is identified. This indicates cutting conditions outside the range of normalcy and is treated as an anomalous state. The cutting task state in an image is identified by its associated dominant element as summarized in Table II. When the dominant element in the image frame is the torch flame (TF), the nominal task is to calibrate the vision system; when it is the preheat pool (PP), to condition the surface; and when it is the combustion pool (CP), to control the torch motion. When no dominant element is found (NA), the corresponding task is to halt cutting operations for inspection and response. In addition, a mismatch between the detected element (perceived task state) and the robot's intended action (internal task state) constitutes anomalous behavior. Sample data for each class is also shown in Fig. 5. As such, the problem is structured as a multinomial classification with four labels: $\{\texttt{TF}, \texttt{PP}, \texttt{CP}, \texttt{NA}\}$. In our dataset, each image is labeled with one of these four labels according to the prominent element in the image and its associated task state.

V. Model Design

In this section, we detail the architecture of our CNN-based model and its input data's preprocessing.

A. Model Architecture

The purpose of the model is to classify input images into one of the aforementioned four cutting task states. The subject and contents of the image data incorporate a large variety of geometrical and chromatic interrelationships between the features of the torch flame, the heat pool, and the surrounding environment. Accordingly, it is non-trivial to express its intricate feature hierarchy via explicit feature engineering. When considering the various model choices for image classification [26] and their respective advantages, we find that CNN models are adequate for our problem given their established capabilities for feature engineering.

Additionally, given our constraints on inference speed for online monitoring requirements, we consider the effects of design parameters on neural network expressivity and performance [27], [28]. In particular, we desire a CNN architecture that is large enough for enabling the model to express a sufficient amount of variations within the data and thus yield a satisfactory prediction accuracy. At the same time, the model's architecture must be small enough for rapid inference during online monitoring.

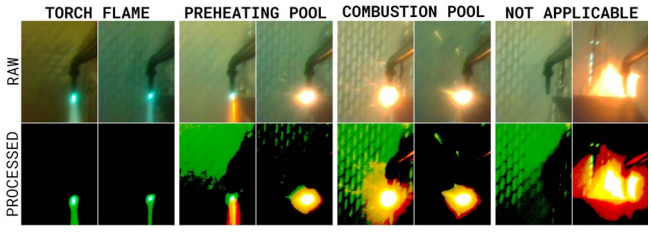

Fig. 6. The preprocessing is demonstrated on samples from each class.

By factoring this tradeoff between prediction accuracy and inference speed during design iterations, we adopt the architecture shown in Fig. 7. Our neural network model $\hat{\mathbf{y}} = f(\mathbf{X}, \theta)$ maps the input $\mathbf{X} \in \mathbb{R}^{3\times 640\times 480}$ which are 3-channel RGB images of size 640×480, to the output $\hat{\mathbf{y}} \in \mathbb{R}^4$ which are the class scores, given the mapping parameters θ, *i.e.*, the model weights. Our network f (expressed in functional form) is composed of four functional blocks,

$$f = \mathsf{B}_4 \circ \mathsf{B}_3 \circ \mathsf{B}_2 \circ \mathsf{B}_1. \tag{1}$$

At the input, we begin with two convolutional functional blocks B_1 and B_2 composed as follows:

$$\begin{aligned} \mathsf{B}_1 &= \texttt{Pool}_{11} \circ \texttt{Drop}_{11} \circ \sigma \circ \texttt{Conv}_{12} \circ \sigma \circ \texttt{Conv}_{11} \\ \mathsf{B}_2 &= \texttt{Pool}_{21} \circ \texttt{Drop}_{21} \circ \sigma \circ \texttt{Conv}_{22} \circ \sigma \circ \texttt{Conv}_{21} \end{aligned} \tag{2}$$

Here, Conv are convolutional layers, Pool are max-pooling layers, Drop are dropout layers (for regularization), σ is the Sigmoid Linear Unit (SiLU) activation function, and the operator $\circ$ is function composition. The blocks B_1 and B_2 are followed by a fully-connected functional block B_3,

$$\begin{aligned} \mathsf{B}_3 = \texttt{Drop}_{33} \circ \sigma \circ \texttt{Dense}_{33} \circ \texttt{Drop}_{32} \circ \sigma \circ \texttt{Dense}_{32} \circ \\ \texttt{Drop}_{31} \circ \sigma \circ \texttt{Dense}_{31} \circ \texttt{Vec} \end{aligned} \tag{3}$$

where Vec vectorizes its input into a 1D vector, and Dense are fully-connected layers. Finally, the output block $\mathsf{B}_4 = \texttt{Dense}_{41}$ contains a single dense layer. The hyperparameters of each block's layers as well as the transformation of the data along the architecture are indicated in Fig. 7.

B. Data Preprocessing

Within our image data, the regions that are more pertinent and crucial are those containing information about the torch flame or heat pool. Accordingly, regions within the image that are relatively brighter are of higher interest, whereas those that are relatively dimmer contain information that is less relevant to our task, *i.e.*, noise. For these reasons, we process the input images (Fig. 6) using thresholding to nullify regions of low interest. At the same time, the thresholding must preserve the desirable bright regions in the image. For

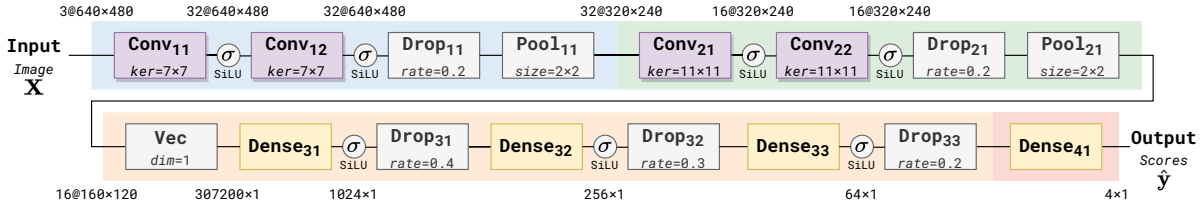

Fig. 7. The architecture of our neural network. Our model is composed of two convolutional blocks (blue and green), one dense block (orange), and one output block (red). The respective hyperparameters of each block are indicated therein and the state of the data (channels, rows, and columns) are displayed along its mappings between layers. The stride of all convolutional layers is $(1, 1)$. All activations are Sigmoid Linear Unit functions.

instance, binary-thresholding (resulting in a binary image) would lose much of the essential information in the bright regions required to reliably discriminate between scenarios. As such, we adopt a one-sided channel-wise thresholding method as specified in the procedure `ProcessImage` below.

define `ProcessImage`(image $\mathbf{X}$):
  $(\alpha_{\mathsf{R}}, \alpha_{\mathsf{G}}, \alpha_{\mathsf{B}}) \leftarrow (175, 150, 225)$
  for each channel $c \in \{\mathsf{R}, \mathsf{G}, \mathsf{B}\}$ **do**
    for each pixel $(i, j) \in \mathbf{X}_c$ **do**
      if $\mathbf{X}_c[i, j] < \alpha_c$ **then** $\mathbf{X}_c[i, j] \leftarrow 0$
  return $\mathbf{X}$

In each individual channel $c \in \{\mathsf{R}, \mathsf{G}, \mathsf{B}\}$, the pixel values are zeroed when they fall below the thresholds α_c, and unchanged otherwise. By performing exploratory data analysis on the training data, we determine the threshold values $(175, 150, 225)$ to sufficiently eliminate noise while preserving relevant information. We note that thresholds are applied channel-wise since the proportion of noise is different across channels, motivating distinct values for α_c.

VI. Training & Evaluation

We delineate our model's training and evaluation below.

A. Model Training and Data Augmentation

Our dataset comprises of 50 experiments recorded at different times-of-day. We split our data into training and testing sets experiment-wise, *i.e.*, we consider 38 recorded experiments for training and 12 for testing, yielding a train-test split of around $(78\%, 22\%)$ or $(111745, 30926)$ images. Through this mutually-exclusive sampling from experiments, we emphasize the learning of key features in the torch flame and heat pool across all experiments and discourage the model from learning the noise, conditions, and background effects unique to each experiment. Additionally for training, we sample our experiment sets across the range of times-of-day to train the model on varied ambient conditions (*e.g.*, lighting, temperature) that are affected by the time-of-day. To further increase our model's robustness against noise in our dataset, we apply several randomized techniques for augmenting our training data. Specifically, we introduce subtle variations in the images by randomizing the following image parameters: shearing, brightness, contrast, saturation, and Gaussian blurring. During training, we minimize the empirical risk as in $\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_k^N L[\mathbf{y}_k, f(\mathbf{X}_k, \boldsymbol{\theta})]$ where L is the categorical cross-entropy loss function, N is the number of training samples, $\mathbf{X}_k$ is the k^{th} training input, and $\mathbf{y}_k$ is its true label. While there are many suitable optimization methods [29] for this minimization problem, we use the adaptive moment estimation (Adam) algorithm [30].

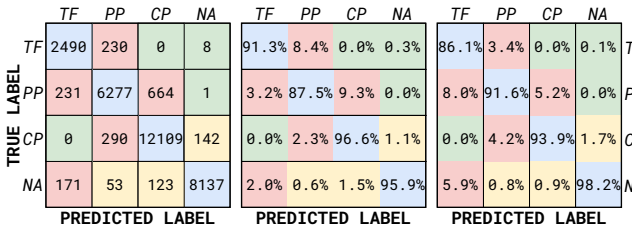

Fig. 8. Confusion matrices of the model's prediction on the test dataset. The diagonals indicate the correct predictions whereas the off-diagonal terms indicate misclassifications. Red, yellow, and green cells indicate label-wise concern that is major, minor, and negligible respectively. **Left:** The matrix is expressed with absolute count. **Center:** The matrix is expressed relative to the true label totals (row-by-row), thereby showing class recall on the diagonal. **Right:** The matrix is expressed relative to the predicted label totals (column-by-column), thereby showing class precision on the diagonal.

B. Model Testing and Evaluation

After testing on the 12-experiment test set, we evaluate our model to assess its output's validity and coherence with our modeling assumptions. Our evaluation metrics for this state classification task are overall model accuracy (Table III) both on the entire test set and its individual experiments, as well as the confusion matrices (Fig. 8) showing class-wise prediction count, recall, and precision. We thus assess our assumption that the task states $\{\mathtt{TF}, \mathtt{PP}, \mathtt{CP}, \mathtt{NA}\}$ are visually distinguishable in our dataset, which would be reflected as sufficiently large diagonal terms in the confusion matrix.

C. Results and Discussion

In general, our model performs reliably, achieving an overall accuracy of 93.814% (Table III). Moreover, its accuracy on individual test sets ranges from 88.725% and above, and lies within $[90.574\%, 96.665\%]$ for two-thirds ($^8/_{12}$) of them. In particular, it distinguishes the `NA` class quite well (with a precision of 98.2% and a recall of 95.9%) despite its intra-class variation (see Fig. 5) suggesting that it can confidently identify anomalous conditions during cutting.

We study the misclassfications (Fig. 8) of concern deemed major (red, 2% or more), minor (yellow, within $[0.5\%, 2\%)$), and negligible (green, less than 0.5%) expressed relative to the true label totals (center matrix) and predicted label totals (right matrix). Major concern occurs mainly during inter-class transitions, for instance, when the cutting task

TABLE III

EVALUATION RESULTS SHOWING THE MODEL'S OVERALL ACCURACY AND CLASS-WISE RECALL AGAINST THE TESTING DATA

Test Set	Total Inputs	Correct Outputs	Model Accuracy	TF Recall	PP Recall	CP Recall	NA Recall
S13	312	312	100.00	—	—	—	100.00
S19	4806	4569	95.069	53.846	100.00	99.242	96.613
S21	2482	2365	95.286	97.541	95.602	94.953	95.210
S25	2909	2581	88.725	100.00	68.916	87.520	97.978
S29	1509	1498	99.271	100.00	—	—	99.265
S34	2472	2239	90.574	97.020	77.215	98.535	94.754
S36	2984	2716	91.019	99.539	78.980	97.853	98.361
S38	2321	2060	88.755	95.556	73.849	97.306	90.022
S40	5277	5101	96.665	97.802	99.208	96.097	92.745
S45	2682	2586	96.421	99.689	96.712	98.419	91.605
S47	1532	1426	93.081	97.458	79.076	98.000	97.073
S49	1640	1560	95.122	98.537	90.864	98.792	95.943
OVERALL	30926	29013	93.814	91.276	87.509	96.555	95.910

Note: Class recalls marked with "—" indicate no data instances of the corresponding class in the particular experiment set.

transitions from having a torch flame (TF) to preheating (PP), or from preheating (PP) to combustion (CP). In addition, the NA class is concentrated before ignition (no torch flame) and after combustion (the heat pool and torch flame gradually extinguish) wherein no dominant element can be identified; these constitute transitions from NA to TF and from CP to NA. Confusion during transitions is reasonable since even human experts struggle to distinguish these transitional images. More so, our results suggest that each transition should have its own respective class and thus more refined labeling.

For assessing inference speed, we evaluated our model on the test set using two representative types of hardware: a special-purpose GPU (NVIDIA A100 Tensor Core) and a mid-range consumer CPU (Intel Core i5-5257U). These respectively represent optimistic and pessimistic estimates of the inference speed, depending on the deployed model's computing hardware. We compute the average time of inferring one image at a time across all test images, resulting in an average inference time of $1.46\ \mathrm{ms}$ (GPU) and $1.25\ \mathrm{s}$ (CPU). In either case, the model's inference is sufficiently fast for prompt anomaly detection and safety response.

VII. CONCLUSION

In this work, we develop a task state classifier for improving the safety of automated oxy-fuel cutting. For this, we curate a labeled dataset by conducting automated cutting experiments using a 1-DOF robot. Our CNN-based model is composed of four functional blocks (each containing its own layers): two convolutional blocks, one dense block, and one dense output block. We preprocess the inputs using one-sided channel-wise thresholding to eliminate noise and preserve the desirable image contents. We train and evaluate our model and achieve an overall accuracy of 93.814% with sufficient average inference speed on both a high-end GPU ($1.46\ \mathrm{ms}$) and a mid-range CPU ($1.25\ \mathrm{s}$). Nevertheless, our model struggles with inter-class transitions motivating the need for more refined classes in future work.

REFERENCES

[1] J. Tilley, "Automation, robotics, and the factory of the future," *McKinsey & Company*, June 2017.

[2] T. Missala, "Paradigms and safety requirements for a new generation of workplace equipment," *Int. J. Occup. Saf. Ergo.*, vol. 20, no. 2, pp. 249–256, 2014.

[3] M. Vagaš, D. Šimšik, A. Galajdová, and D. Onofrejová, "Safety as necessary aspect of automated systems," in *Int. Conf. on Emerging eLearning Technologies and Applications*, 2018, pp. 617–622.

[4] K. Nachbargauer, "Oxy-fuel cutting: Automation makes the difference," *Zavarivanje i zavarene konstrukcije*, vol. 64, no. 1, pp. 39–45, 2019.

[5] J. Smart, G. Lu, Y. Yan, and G. Riley, "Characterisation of an oxy-coal flame through digital imaging," *Combust. Flame*, vol. 157, no. 6, pp. 1132–1139, 2010.

[6] G. Lu, G. Gilabert, and Y. Yan, "Vision based monitoring and characterisation of combustion flames," *J. Phys: Conf. Ser.*, vol. 15, no. 1, p. 194, Jan 2005.

[7] R. Sekhar, D. Sharma, and P. Shah, "Intelligent classification of tungsten inert gas welding defects: A transfer learning approach," *Front. Mech. Eng.*, vol. 8, 2022.

[8] Z. Wang, H. Chen, Q. Zhong, S. Lin, J. Wu, M. Xu, and Q. Zhang, "Recognition of penetration state in GTAW based on vision transformer using weld pool image," *Int. J. Adv. Manuf. Technol.*, vol. 119, no. 7-8, pp. 5439–5452, 2022.

[9] C. Li, Q. Wang, W. Jiao, M. Johnson, and Y. M. Zhang, "Deep learning-based detection of penetration from weld pool reflection images," *Weld. J.*, vol. 99, no. 9, pp. 239s–245s, 2020.

[10] W. Jiao, Q. Wang, Y. Cheng, R. Yu, and Y. Zhang, "Prediction of weld penetration using dynamic weld pool arc images," *Weld. J.*, vol. 99, pp. 295s–302s, 2020.

[11] Y. Feng, Z. Chen, D. Wang, J. Chen, and Z. Feng, "DeepWelding: a deep learning enhanced approach to GTAW using multisource sensing images," *IEEE Trans. Ind. Inf.*, vol. 16, no. 1, pp. 465–474, 2020.

[12] K. Zhu, W. Chen, Z. Hou, Q. Wang, and H. Chen, "F2GAN based few shot image generation for GMAW defects detection using multi-sensor monitoring system," 2022.

[13] M. Rohe, B. N. Stoll, J. Hildebrand, J. Reimann, and J. P. Bergmann, "Detecting process anomalies in the GMAW process by acoustic sensing with a convolutional neural network (CNN) for classification," *J. Manuf. Mater. Process.*, vol. 5, no. 4, 2021.

[14] R. S. Barot and V. J. Patel, "Process monitoring and internet of things feasibility for submerged arc welding: State of art," *Mater. Today:. Proc.*, vol. 45, pp. 4441–4446, 2021.

[15] D. Wu, H. Chen, Y. Huang, and S. Chen, "Online monitoring and model-free adaptive control of weld penetration in VPPAW based on extreme learning machine," *IEEE Trans. Ind. Inf.*, vol. 15, no. 5, pp. 2732–2740, 2019.

[16] S. Kang, M. Kang, Y. H. Jang, and C. Kim, "Deep learning-based penetration depth prediction in Al/Cu laser welding using spectrometer signal and CCD image," *J. Laser Appl.*, vol. 34, no. 4, p. 42035, 2022.

[17] S. Oh, H. Kim, K. Nam, and H. Ki, "Deep-learning approach for predicting laser-beam absorptance in full-penetration laser keyhole welding," *Opt. Express*, vol. 29, no. 13, pp. 20 010–20 021, Jun 2021.

[18] S. Shevchik, T. Le-Quang, B. Meylan, F. V. Farahani, M. P. Olbinado, A. Rack, G. Masinelli, C. Leinenbach, and K. Wasmer, "Supervised deep learning for real-time quality monitoring of laser welding with X-ray radiographic guidance," *Sci. Rep.*, vol. 10, no. 1, p. 3389, 2020.

[19] B. Shen, J. Lu, Y. Wang, D. Chen, J. Han, Y. Zhang, and Z. Zhao, "Multimodal-based weld reinforcement monitoring system for wire arc

additive manufacturing," *J. Mater. Res. Technol.*, vol. 20, pp. 561–571, 2022.

[20] C. Xia, Z. Pan, Y. Li, J. Chen, and H. Li, "Vision-based melt pool monitoring for wire-arc additive manufacturing using deep learning method," *Int. J. Adv. Manuf. Technol.*, vol. 120, no. 1-2, pp. 551–562, 2022.

[21] N. D. Jamnikar, S. Liu, C. A. Brice, and X. Zhang, "Comprehensive process-molten pool relations modeling using CNN for wire-feed laser additive manufacturing," *ArXiv*, vol. abs/2103.1, 2021.

[22] D. K. Maxime, H. N. Raymond, P. Olivier, and B. Tibi, "Anomaly detection in orthogonal metal cutting based on autoencoder method," in *Int. Conf. on Intelligent Systems*, 2018, pp. 485–493.

[23] C. Ajmi, J. Zapata, S. Elferchichi, A. Zaafouri, and K. Laabidi, "Deep learning technology for weld defects classification based on transfer learning and activation features," *Adv. Mater. Sci. Eng.*, vol. 2020, p. 1574350, 2020.

[24] M. Sun, M. Yang, B. Wang, L. Qian, and Y. Hong, "Applications of molten pool visual sensing and machine learning in welding quality monitoring," *J. Phys: Conf. Ser.*, vol. 2002, no. 1, p. 12016, Aug 2021.

[25] J. Rojas, Z. Huang, and K. Harada, "Robot contact task state estimation via position-based action grammars," in *IEEE International Conference on Humanoid Robots*, 2016, pp. 249–255.

[26] P. Wang, E. Fan, and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," *Pattern Recognit. Lett.*, vol. 141, pp. 61–67, 2021.

[27] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein, "On the expressive power of deep neural networks," in *Int. Conf. Mach. Learn.* PMLR, 2017, pp. 2847–2854.

[28] A. Elhassouny and F. Smarandache, "Trends in deep convolutional neural networks architectures: a review," in *2019 Int. Conf. Comput. Sci. Renew. Energies*, 2019, pp. 1–8.

[29] S. Sun, Z. Cao, H. Zhu, and J. Zhao, "A survey of optimization methods from a machine learning perspective," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3668–3681, 2020.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd Int. Conf. Learn. Represent.*, 2015.