information

MDPI

Article

Feature Extraction of Laser Machining Data by Using Deep Multi-Task Learning

Quexuan Zhang *, Zexuan Wang †, Bin Wang †, Yukio Ohsawa and Teruaki Hayashi

Department of Systems Innovation, School of Engineering, The University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656, Japan; wangzexuan@g.ecc.u-tokyo.ac.jp (Z.W.); binw@g.ecc.u-tokyo.ac.jp (B.W.); ohsawa@sys.t.u-tokyo.ac.jp (Y.O.); hayashi@sys.t.u-tokyo.ac.jp (T.H.)

* Correspondence: hinson@g.ecc.u-tokyo.ac.jp

† These authors contributed equally to this work.

Received: 30 June 2020; Accepted: 23 July 2020; Published: 27 July 2020

Abstract: Laser machining has been widely used for materials processing, while the inherent complex physical process is rather difficult to be modeled and computed with analytical formulations. Through attending a workshop on discovering the value of laser machining data, we are profoundly motivated by the recent work by Tani et al., who proposed in situ monitoring of laser processing assisted by neural networks. In this paper, we propose an application of deep learning in extracting representative features from laser processing images with a multi-task loss that consists of cross-entropy loss and logarithmic smooth L^1 loss. In the experiment, AlexNet with multi-task learning proves to be better than deeper models. This framework of deep feature extraction also has tremendous potential to solve more laser machining problems in the future.

Keywords: deep learning; feature extraction; multi-task learning; laser processing

1. Introduction

The application of deep learning methods to physics is gaining increasing attention due to its powerful ability in modeling and predicting. Laser machining has highly reformed the manufacturing industry over recent decades, and it has also become a popular topic in the field of physical studies. However, the complex nonlinear process inherent to laser processing is still a problem which remains. In this paper, we demonstrate an application of deep learning in extracting representative features from laser processing images with a multi-task learning scheme.

1.1. Laser Machining

Laser machining is a physical process of removing material via the interaction between a laser beam and some target material. In laser machining processes, the energy of a photon is transported to the target material in the form of thermal energy or photochemical energy, and then the target material is removed by melting or ablation [1]. Laser machining has been characterized by a lot of advantages such as flexibility, precision, automation, and versatility [2]. It has been widely applied to high-precision materials processing in recent years. The global laser machining market is expected to reach USD 5.7 billion by 2022 due to the increasing need for high-precision and automation in manufacturing [3]. Laser machining is believed to play an important role in Society 5.0 [4].

Nevertheless, it is of great difficulty to strictly control the machining quality due to the inherent complex physical process. Sometimes a slight change in the laser or environmental parameters could lead to a totally different result [5].

Since it is as yet impracticable to simulate this complex process by mathematical or physical methods, deep learning methods as pattern recognition algorithms are drawing more attention recently.

Deep learning approaches allow stakeholders to skip obtaining complex prior physical knowledge of laser machining. This also helps users with better mining the value of machining data from another perspective.

1.2. Purpose and Motivation

Although there is huge potential for applying deep learning in physics, the corporation between data science and physics is sometimes hard to achieve due to the lack of cross-disciplinary communication. However, our work is strongly motivated by participating an IMDJ workshop [6] to seek solutions to handle a series of problems in physics, including laser machining.

Here, IMDJ is a game-style workshop to discover the value of data and find solutions for some practical problems, in the way of creating new ideas by combining Data Jackets (DJ) and Tool Jackets (TJ) with negotiations. A DJ keeps the digest of a dataset in a structured format so that the dataset could be comprehended in the discussions without showing the actual content [7]. Similarly, a TJ is a summary of a certain technical tool that might be complicated for non-experts in data utilization. Besides, a visualization method called KeyGraph [8] is usually used to reveal the relationships between different DJs and TJs, which could make it easier for the participants to discuss on cross-disciplinary data or techniques and then create new solutions with them.

Many influential data scientists and physicists from The University of Tokyo attended the IMDJ workshop in which we participated. This workshop aimed at using methods from data science to solve problems which were still complex in physics. The physicists proposed their datasets and the requirements on them while data scientists introduced data utilization methods for possible solutions. In this way, a cross-disciplinary collaboration could be built without requiring prior knowledge of other fields for all participants. Many latent problems and applications in the fields of physics and data science were deeply discussed in this workshop. From the discussions, we are highly motivated by methods put forward by some physicists of using deep learning on laser machining data. Especially, Tani, Aoyagi, and Kobayashi [9–11] recently proposed in situ process monitoring assisted by a deep neural network, which does not require analytical formulation (see also Section 2.1). This gave us the inspiration for this work to apply deep learning methods for the feature extraction of laser machining data. Because it is difficult for humans to extract useful information from the laser machining data where the speckle patterns are captured on the Fourier plane, we considered that deep learning techniques could be fully utilized to reveal more essential in-data information [12,13].

The main contribution of this paper is that we analyze the laser machining data which is still less studied, and then design a deep multi-task learning framework to train a feature-extracting model for the downstream tasks with the help of some known information, such as processing power settings or logarithmic orders of machining stages. Besides, we will demonstrate that AlexNet with multi-task performs better than a deeper model, which could also meet the real-time requirements due to the less computational cost.

2. Related Work

2.1. Laser Machining and Deep Learning

Recently there has been some research for applying deep learning methods on laser machining data. The pioneering work proposed by Tani et al. [11] introduced a method to monitor the progress of laser processing using laser speckle patterns without a need for analytical formulation. Deep learning methods were used to extract multiple information such as ablation depth and material type under processing, which could be useful for composite material processing. Their work proved the simplicity, versatility and accuracy of applying deep learning in laser processing. Another deep learning-based method was proposed by Mills et al. [14] as image-based monitoring of femtosecond laser machining. This paper aimed to build a real-time feedback system in laser machining by predicting the type of material, the laser fluence, and the number of pulses at the same time as a classification problem.

The disadvantage of this method is that the environmental parameters were strictly limited since the training set only contains a small number of all possible combinations.

Existing work focuses less on the aspect of feature extraction than just taking deep learning as pattern recognition or regression algorithms. With feature extraction, our results could be extended and used in more potential physical problems of laser machining.

2.2. Multi-Task Learning

The traditional solution to obtain machine learning models for different tasks on the same dataset, or the same task on a different dataset is to train different new models from scratch each time. However, in some real-world applications such as medical image analysis or high precision physical experiments, enough data samples in good quality are often difficult to collect. In this case, training models separately with limited data may lead to a result of several low-accuracy shallow models, which is not desirable in real applications.

Multi-task learning (MTL) [15] is inspired by human learning activities when people often tend to apply the knowledge obtained from previous tasks to help with working on a new but related task. It is considered a good solution when there are multiple related tasks and each of which only has limited training samples. Among these learning tasks, all of them are assumed to be related to each other. In this case, it can be found that learning these tasks jointly could lead to a performance improvement compared with learning them separately. MTL has seen a lot of success across many applications of machine learning, such as natural language processing [16], speech recognition [17] and computer vision [18].

Deep MTL [19,20] combines deep learning and MTL where multiple learning tasks will be solved simultaneously by exploring commonalities and differences among all the tasks by leveraging deep neural networks. Recently, deep MTL has started to draw scholars' attention due to its capacity of learning hierarchical features and sharing knowledge from different domains. One of the reasons for the success of deep MTL could be attributed to the inbuilt sharing system, which allows a network to extract features shared across different tasks [21,22].

In this paper, a deep multi-task learning framework will be applied for better mining latent information in the laser machining data through a composed loss function.

3. Dataset

In this section, we describe the laser machining image dataset we obtained by negotiating with the physicists in the IMDJ workshop. We introduce the data details and then report an exploratory analysis on a part of the dataset.

3.1. Details

The laser machining dataset adopted in this paper is kindly provided by one of the workshop participants, Kobayashi Lab., Institute for Solid State Physics (ISSP), The University of Tokyo. There are 10 different laser power settings, 105 independent experiments in each power setting, and 250 sequential stages within each experiment so that there are a total of 262,500 images in the dataset. Each image records speckle patterns in the Fourier transform plane with a resolution of 400 × 4080 pixels in grayscale, and all the images have already been labeled with laser powers and stage numbers. For the aim of utilizing machine learning methods and the consideration of experimental reproduction, the total dataset is further divided into three subsets shown in Table 1.

Table 1. The data subsets for our tasks.

Subset Name	Range of Experiment IDs	Total Number of Samples
Training	1–70	175,000
Validation	71–85	37,500
Test	86–105	50,000

Nevertheless, the size of the original images is too large that they exceed the memory limitation of our device, so we use bilinear interpolation to resize the input images from the original 400 × 4080 pixels to a smaller size. We choose 224 × 224 as the new size because it is mostly used by existing CNN models. This preprocessing step also accelerates the training speed. Moreover, each data value in the three sets is normalized by the empirical mean 0.109251 and standard deviation 0.033309 which are observed over the training set.

3.2. Analysis

To understand the laser processing image data, we give a study looking at the training set by using principal component analysis (PCA) [23]. PCA is one of the methods that explore the characteristics of data sets by finding orthogonal components, on which the projection of the data has the most variance. Before performing PCA, the original data is usually processed with mean centering that subtracts each data value from the empirical mean along each variable.

Consider a centered real matrix $\boldsymbol{X}$ of $N \times M$ size, where N is the number of samples, M is the number of variables of the data and $N \geq M$. PCA performs eigenvalue decomposition on the covariance matrix $\boldsymbol{C} = \boldsymbol{X}^\top \boldsymbol{X}/(N-1)$ to find eigenvectors as the components with the largest-to-smallest sorted eigenvalues $\lambda_{[M]}(\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M)$. The eigenvectors and the corresponding eigenvalues are used to explain the variance in the data with explained variance ratio $r_i = \lambda_i / \sum_{j=1}^{M} \lambda_j$. However, it is hard for us to operate eigenvalue decomposition on our training set directly where $N = 175{,}000$ and $M = 224 \times 224 = 50{,}176$. Therefore, we apply singular value decomposition (SVD) on the centered training set alternatively. SVD gives $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices, and $\boldsymbol{\Sigma}$ is an $M \times M$ diagonal matrix of singular values $\sigma_{[M]}$. In practice, the computation can drop the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, and only store the diagonal matrix $\boldsymbol{\Sigma}$ as an array with M size. Letting the singular values be in the order $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M$, we can obtain the eigenvalues of $\boldsymbol{C}$ by

$$\lambda_i = \frac{\sigma_i^2}{N-1}. \tag{1}$$

To evaluate the amount of variance explained by the components, we use the cumulative explained variance ratio (CEVR) which is defined as

$$R_i = \frac{\sum_{j=1}^{i} \lambda_j}{\sum_{k=1}^{M} \lambda_k}. \tag{2}$$

Besides mean centering, we handle the centered training set before performing SVD with normalization that divides the data value by the empirical standard deviation over each variable. The data processed with the combination of centering and normalization are called z-scores, which could improve the performance in some machine learning methods. Then we obtain the singular values of the z-scores with SVD and calculate the CEVRs by Equations (1) and (2) for each component. The result is shown in Figure 1.

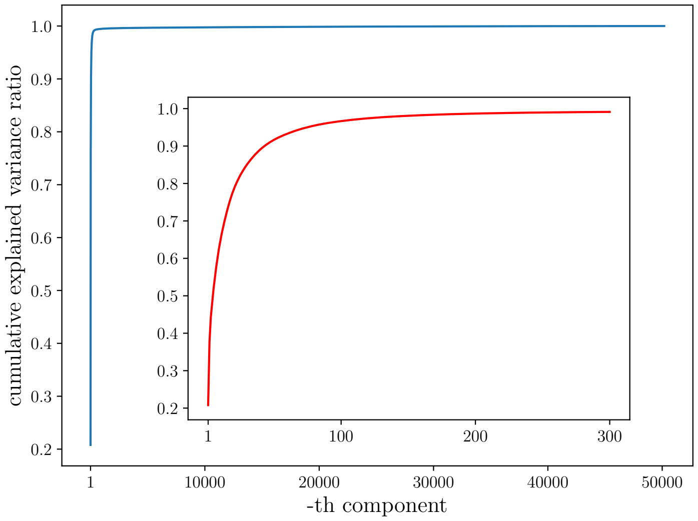


Figure 1. CEVR R_i obtained by using SVD. The blue line shows for all components in the training set, while the red one only shows for $1 \leq i \leq 300$.

Although a 224 × 224 image has a high dimension, we find that we can adopt less than 300 components to recover greater than 99% variance in the training data. This can help us extract features with a low-rank decomposition, e.g., truncated SVD [24], for alleviating the curse of dimensionality. Matrix decomposition is widely used in traditional machine learning methods for high-dimensional data.

4. Method

To utilize the speckle pattern image data for the downstream applications such as ablation prediction or laser machining monitoring, extracting features from the original data is the most critical step, as any further applications will be based on the representative features extracted from the images. A good form of image feature representations will increase the accuracy performance of the model and the ability of the model to be applied to more datasets in the future.

In this paper, we adopt two CNN models for feature extraction and design two corresponding tasks to evaluate the performance of feature extraction on speckle pattern data:

1. Power Classification: Input an image, then predict the corresponding laser source power setting when this image was taken, i.e., classify the image to one of the 10 classes of laser power.
2. Shot No. Regression: Input an image, then predict the logarithmic corresponding shot no. of this image, i.e., at which stage during a single experiment this image was taken, the shot no. can be one value in the range of 1–250.

To handle these two tasks, three steps are designed in our proposed method including image feature extraction with CNN, classification and regression, and MTL (Figure 2). The details of these steps will be introduced in the following sections.

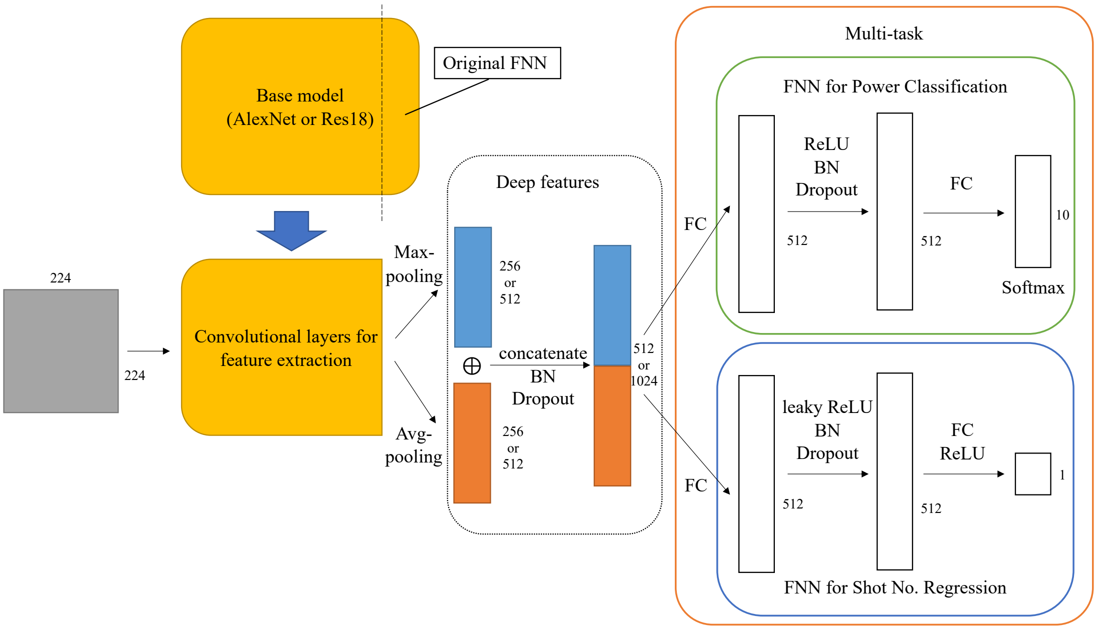


Figure 2. The diagram of the architecture in our method. The numbers are the sizes of their nearest sides while the default is 1.

4.1. Image Feature Extraction with CNN

The first as well as the most important step of our proposed model is to extract the features of images from the original data so that we can use these features to represent images with similar structures, which are more likely to be taken in similar laser source settings or experiment stages. In this step, we adopt two widely used CNN models in the field of computer vision, AlexNet and ResNet, for image feature extraction as our base models:

- AlexNet [25] has five convolutional layers and three fully-connected (FC) layers and uses the rectified linear unit (ReLU) as the activation function instead of the sigmoid function to reduce gradient vanishing and gradient exploding problems. AlexNet also introduces mechanisms such as Dropout and overlapping pooling to avoid overfitting.
- ResNet [26] (Deep Residual Network) is designed for networks with great depths by introducing a new neural network layer, Residual Block, to alleviate the problem of training very deep networks. The most widely used variances of ResNet include Res18, which has 17 convolutional layers and one fully-connected layer.

For feature extraction, we drop all the original fully-connected layers at the ends of the two base models, connect the last convolutional layers to parallel average- and maximum-pooling layers, and concatenate the two pooling results to a vector as the extracted features. Furthermore, both pooling layers operate over the neurons on each filter of the last convolutional layer, so that the number of the extracted features is two times the number of the channels of the last convolutional layer. Accordingly, the sizes of deep features extracted by AlexNet and Res18 are 512 and 1024 respectively. We expect that average-pooling could transfer the overall extracted information while maximum-pooling could select significant features.

In general comparison, ResNet has a better performance in accuracy than AlexNet. However, it shows different results on the laser machining data according to the experiments mentioned later.

4.2. Classification and Regression

In this step, we design different loss functions for the models solving the two tasks introduced above. Loss function evaluates the error between real label value and predicted label value output by the model, i.e., a minimized loss value tends to imply that the model fits the dataset well, which is also the goal of the training of a learning model. However, for different tasks conducted on the same dataset, we may focus on different domains of the data, therefore, to fulfill different task requirements, we need to design different loss functions accordingly.

For Power Classification, we can directly use the cross-entropy loss function as this is a classic multi-classification problem, which can be denoted as

$$\mathcal{L}_p = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k} \log p_{i,k}, \tag{3}$$

where for the i-th input sample (x_i, y_i), c_i is the labeled laser source power setting of an image x_i, and $y_{i,k}$ is the one-hot representation of the sample label where

$$y_{i,k} = \begin{cases} 1, & \text{if } c_i = k \\ 0, & otherwise \end{cases}. \tag{4}$$

The probability of the i-th sample will be predicted with label k is $p_{i,k}$, and there are K labels (in this study, according to the dataset, $K = 10$) and N samples in total.

For Shot No. Regression, we employ smooth L^1 loss [18] to build a regression model on the shot no. of each image. The reason why we use smooth L^1 loss instead of normal L^1 loss or squared L^2-norm is that it could avoid propagating too large gradients when the absolute loss is greater than 1; also, it could do soft learning when the loss is in the range $[-1, 1]$. Another modification is that because a shot no. is just a discrete integer in the specified interval, we could then apply a logarithmic function to it to relax this strong constraint. Therefore, the loss can be denoted as

$$\mathcal{L}_s = \frac{1}{N}\sum_{i=1}^{N} \text{smooth-}L^1\left(z_i - z'_i\right), \tag{5}$$

where for the i-th input sample (x_i, z_i), z_i is the logarithmic real shot no. of image x_i, z'_i is the predicted logarithm, and

$$\text{smooth-}L^1(a) = \begin{cases} 0.5a^2, & \text{if } |a| \leq 1 \\ |a| - 0.5, & otherwise \end{cases}. \tag{6}$$

4.3. Multi-Task Learning

For our study, the idea of MTL is adopted as the designed two tasks are considered related and they all serve the same aim to help neural network model to better extract features from the speckle pattern images. Laser ablation shows that depth sequences of laser machining are variant by different power settings, so we believe that speckle patterns in processing stages are relative to the power. By combining the two mentioned losses, we can consider the $\mathcal{L}_s$ as a regularization term to $\mathcal{L}_p$. Furthermore, instead of setting a specific regularization term that encodes the relationship between the two tasks, sharing convolutional layers can reduce the amounts of model parameters for the two tasks. If we encode the relationship between the two tasks, we may increase space and time complexity by the square of the amounts of model parameters, which seems impractical for deep learning methods. We will show that these help us get an overall better performance and generalization than training the two tasks individually by the later experimental results.

By sharing the same neural network layers and combining different loss functions at output layers, the overall loss function can be denoted as

$$\begin{aligned}\mathcal{L} &= \alpha\mathcal{L}_p + (1-\alpha)\mathcal{L}_s \\ &= \frac{1}{N}\sum_{i=1}^{N}\left(-\alpha\sum_{k=1}^{K} y_{i,k}\log p_{i,k} + (1-\alpha)\text{smooth-}L^1\left(z_i - z_i'\right)\right),\end{aligned} \tag{7}$$

where $0 \leq \alpha \leq 1$ is the hyper-parameter to adjust the weight of different loss functions, i.e., the importance of variant tasks. In this study, we fix it to $\alpha = 0.5$, which means these two tasks are treated equally.

5. Results and Discussions

To evaluate the performance of feature extraction on the laser processing data, we introduce several metrics, execute the mentioned deep learning tasks and compare them to the result by neural networks without feature extraction and one of the traditional machine learning methods with generally high performance, support vector machine (SVM) [27].

5.1. Metrics and Settings

In this paper, we introduce accuracy (ACC), precision (PR), recall (RC), F^1 score into the evaluation of Power Classification, and mean absolute error (MAE), R^2 score into the one of Shot No. Regression. For the evaluation over N samples and K classes,

$$ACC = \frac{1}{N}\sum_{i=1}^{N} 1(c_i = c_i'), \tag{8}$$

$$PR_k = \frac{TP_k}{TP_k + FP_k} = \frac{\sum_{i=1}^{N} 1(c_i' = k \wedge c_i = k)}{\sum_{i=1}^{N} 1(c_i' = k)}, \tag{9}$$

$$RC_k = \frac{TP_k}{TP_k + FN_k} = \frac{\sum_{i=1}^{N} 1(c_i' = k \wedge c_i = k)}{\sum_{i=1}^{N} 1(c_i = k)}, \tag{10}$$

$$F_k^1 = \frac{2 \times PR_k \times RC_k}{PR_k + RC_k}, \tag{11}$$

$$PR = \frac{1}{K}\sum_{k=1}^{K} PR_k, \quad RC = \frac{1}{K}\sum_{k=1}^{K} RC_k, \quad F^1 = \frac{1}{K}\sum_{k=1}^{K} F_k^1, \tag{12}$$

where $1(\cdot)$ is the indicator function, c_i' indicate the predictive power of i-th sample, and TP_k, FP_k and FN_k are the numbers of true positives, false positives and false negatives for class k respectively;

$$MAE = \frac{1}{N}\sum_{i=1}^{N} \left|z_i - z_i'\right|, \tag{13}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(z_i - z_i'\right)^2}{\sum_{i=1}^{N}\left(z_i - \frac{1}{N}\sum_{i=1}^{N} z_i\right)^2}. \tag{14}$$

ACC, PR, RC, F^1, and R^2 are the higher the better, while MAE is the lower the better. Because the numbers of samples in each class are the same on our data subsets, $ACC = RC$.

In the experiment, we use the PyTorch [28] implementations of AlexNet and ResNet. The concatenated deep features are passed to batch normalization (BN) layer, and 0.25-Dropout for better stability and generalization. In a classification model, there is a 2-hidden-layer fully-connected neural network (FNN) following the deep feature output. Each hidden layer of the FNN is 512-sized and the first one followed by a ReLU, a BN layer and a 0.5-Dropout sequentially. The neural network used for regression is similar to the one for classification, yet we adopt leaky ReLU with a 0.3 negative slope as the first activation. For MTL, we adjoin these two sorts of neural networks to the feature output layer together. To optimize models' parameters, we employ stochastic gradient descent with weight decay 1.0×10^{-4}. We also apply a triangular cyclic scheduler [29] to adjust the learning rate within the range $[1.0 \times 10^{-3}, 6.0 \times 10^{-2}]$ and the momentum within $[0.8, 0.9]$ by 16.5 epochs for each slope of the triangles. For each epoch, we shuffle the training images with the batch size 256. In addition, we choose the 17 convolutional layers version for ResNet. The models are trained 100 epochs and selected by achieving both the higher accuracy and the lower loss on the validation set.

To compare with the deep learning methods, we also use two other machine learning methods SVM and simple FNN. SVM maps the data to a high- or infinite-dimensional space so that the data points are separate enough to divide to different targets. SVM is usually used with a linear or nonlinear kernel to help the data mapping, but we find that linear SVM performs extremely better than one with generally used Radial Basis Function on our dataset. According to the result of the analysis in Section 3.2, we transform the data to z-scores and reduce the dimension of each sample to 260 by applying truncated SVD with the top 260 singular values, where $R_{260} > 0.99$. Then we train two SVM models for the two tasks on the dimension-reduced training set and test them on the validation and test sets. Besides, For simple FNN, the architectures are the same as the one followed deep features mentioned above, while an input is a 50176-sized vector by flattening a 2-dimensional speckle pattern image.

5.2. Results

In this section, we first compare the performance of different methods by using the metrics mentioned above. Then, we discuss the benefit of MTL against single-task learning (STL) on the dataset.

The results in Tables 2 and 3 shows that AlexNet model with MTL outperforms the others for both tasks, especially the traditional SVM method. Even though the decomposed data retain greater than 0.99 variance ratio, the linear spaces still lack enough connection among the variables to discover good divisions for the problems. When introducing CNN for feature extraction, the neural networks perform better than simply flattening the original image data. FNN-only could not extract features well, albeit with MTL. In the CNN-used cases, ResNet has a deeper architecture, however, it performs worse than AlexNet in our experiment settings in not only the evaluation but also the time and space cost. We consider that the Fourier transform in the data of speckle patterns could be treated as a part of feature extraction layers on the top of the whole model. Because the "parameters" of the Fourier transform can not be tuned during training, the deeper the model is, the harder the parameters of the later layers are optimized. Furthermore, we observe that the smooth L^1 loss is usually smaller than the cross-entropy loss in our settings, so that deeper models gain less backward information in the regression task.

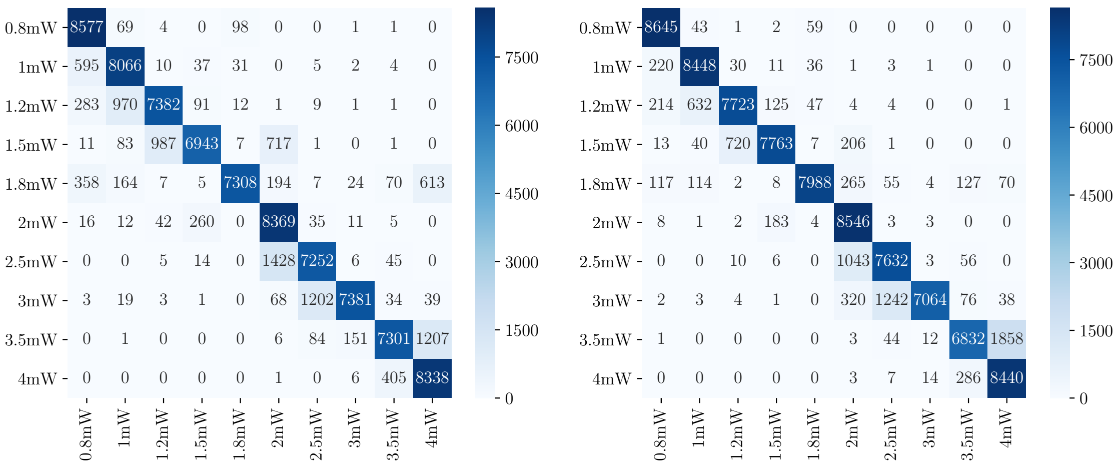
Table 2. The classification result of the tasks with different models.

Model	Validation			Test		
	ACC	PR	F^1	ACC	PR	F^1
SVM with SVD	0.44333	0.49753	0.43899	0.52022	0.53973	0.49488
simple FNN	0.71637	0.70822	0.71340	0.73178	0.77212	0.72773
simple FNN with MTL	0.71803	0.72773	0.70776	0.75502	0.78540	0.74073
AlexNet	0.87184	0.88112	0.87064	0.88446	0.89578	0.88406
AlexNet with MTL	**0.90069**	**0.90809**	**0.90032**	**0.9061**	**0.91547**	**0.90524**
ResNet	0.87912	0.89091	0.87580	0.89204	0.90394	0.89191
ResNet with MTL	0.85171	0.87135	0.85183	0.88202	0.89715	0.87770

Table 3. The regression result of the tasks with different models.

Model	Validation		Test	
	MAE	R^2	*MAE*	R^2
SVM with SVD	0.83891	−0.39754	0.83301	−0.37363
simple FNN	0.40982	0.70053	0.41074	0.70823
simple FNN with MTL	0.42202	0.68666	0.41330	0.69938
AlexNet	0.35468	0.73977	0.37303	0.71816
AlexNet with MTL	**0.28893**	**0.84342**	**0.29558**	**0.82798**
ResNet	0.35415	0.76356	0.37520	0.76356
ResNet with MTL	0.32888	0.79420	0.34177	0.78346

For Power Classification, we construct confusion matrices for the predictive results of classification (Figure 3). In a confusion matrix, each column denotes an actual class while each row is a predicted class. The values located at the diagonal of the matrix are the numbers of corresponding correct predictions, while the others are the ones of corresponding prediction errors. We find that the concentration of the diagonal with MTL is higher than the one with STL. MTL reduces the numbers of the errors for most of the labels, especially for the samples shot by 1.8 mW power. Despite the decline of the numbers of the corrects for 3.0 mW and 3.5 mW, the errors still locate at the neighborhoods mostly.

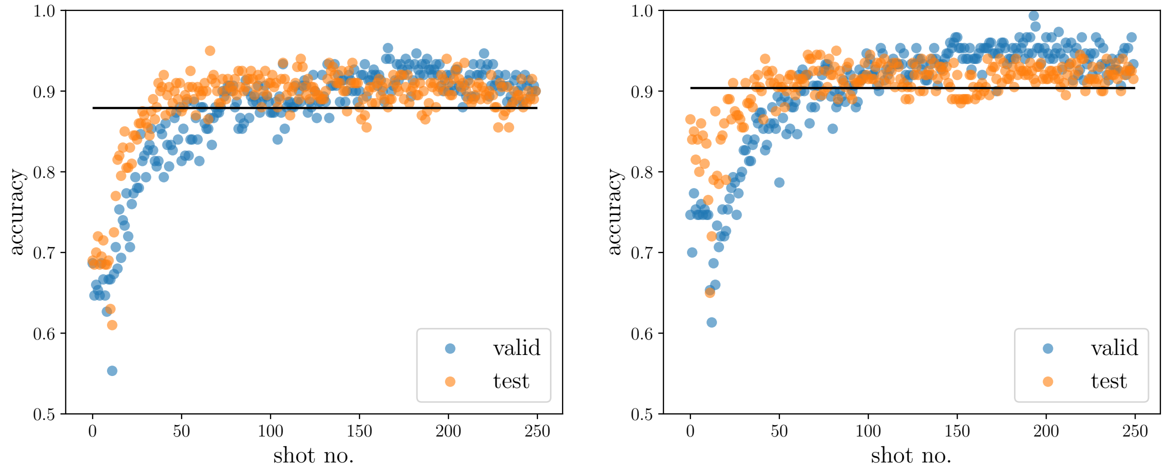


(**a**) STL (**b**) MTL

Figure 3. Confusion matrices of the predictions for Power Classification by AlexNet models with (**a**) STL and (**b**) MTL.

Additionally, we plot *ACC* of Power Classification and *MAE* of Shot No. Regression over each shot no. in Figures 4 and 5. The *p*-values are given by the one-way ANOVA tests. Noticeably, there are watersheds near the twenty-fifth shots because the volume of speckle patterns is too little to generate enough information on the image at the beginning steps of laser processing. Nevertheless, these results show that MTL helps not only the predictions of the later steps but also the ones of the beginnings. The reason is that MTL could make the optimization concern complementary information of power and shot no. simultaneously, while STL has no reference to other information sources.

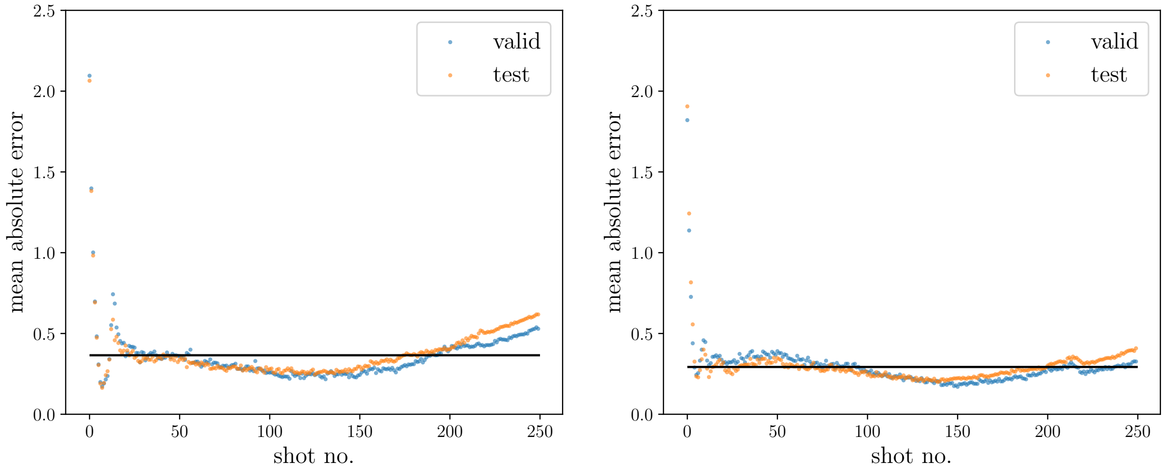
MTL enhances most of the power predictions, but the ones for 3.0 mW and 3.5 mW perform a little badly according to the confusion matrix. Similarly, for Shot No. Regression, the *MAE* of the shots later than about 200th tends to rise. The reason for that may be the anomalous data (e.g., the material has been cracked by a high power or many time shots) affect the training and the predicting, which is the limit of the discriminative models we used. The self-supervised learning approach [30] is a candidate to help us solve this problem and develop the anomaly detection method in future work. Furthermore, we pass shot no. to training models with MTL, yet the models handle only one image at a time in the prediction procedure. Utilizing the time-series information could be another way for us to improve the methodology.



(**a**) STL ($p = 0.024$)

(**b**) MTL ($p = 0.279$)

Figure 4. The comparison of ACC of Power Classification over each shot no. between (**a**) STL and (**b**) MTL is significantly different ($p < 0.001$). The black lines denote the ACC over the validation and test sets.



(**a**) STL ($p = 0.216$)

(**b**) MTL ($p = 0.572$)

Figure 5. The comparison of MAE of Shot No. Regression over each shot no. between (**a**) STL and (**b**) MTL is significantly different as well ($p < 0.001$). The black lines denote the MAE over the validation and test sets.

6. Conclusions

In this paper, we present an application of deep learning for the feature extraction of laser machining data, which is inspired by attending the IMDJ workshop and the work of laser processing monitoring. Through the experiment, we find that AlexNet with multi-task learning performs better than ResNet or single-task model. Because the computational cost of AlexNet is less than ResNet, it could be easier used for real-time applications. We can employ this feature extraction framework to enlarge the use of deep learning for other related laser machining problems, e.g., ablation depth prediction on other materials. However, this method is supervised so that it is dependent on the label information. In the future, we will introduce an unsupervised [31] or self-supervised fashion to mine the data features more deeply.

Author Contributions: All listed authors have contributed substantially to the work, namely: conceptualization, Q.Z.; methodology, Q.Z. and Z.W.; software, Q.Z., Z.W. and B.W.; validation, Q.Z.; formal analysis, Q.Z.; investigation, Q.Z. and B.W.; resources, Y.O.; data curation, Q.Z.; writing—original draft preparation, Q.Z., Z.W., and B.W.; writing—review and editing, Y.O. and T.H.; visualization, Q.Z.; supervision, Y.O.; funding acquisition, Y.O. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the MEXT Quantum Leap Flagship Program, grant number JPMXS0118067246, and JSPS KAKENHI JP19H05577.

Acknowledgments: The authors would like to acknowledge the participants of the workshops, as well as the support of Kobayashi Lab., who provided the necessary resources for the success of the research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chryssolouris, G.; Stavropoulos, P.; Salonitis, K. Process of Laser Machining. In *Handbook of Manufacturing Engineering and Technology*; Nee, A., Ed.; Springer: London, UK, 2013.
2. Tofail, S.A.; Koumoulos, E.P.; Bandyopadhyay, A.; Bose, S.; O'Donoghue, L.; Charitidis, C. Additive manufacturing: Scientific and technological challenges, market uptake and opportunities. *Mater. Today* **2018**, *21*, 22–37. [CrossRef]
3. Shiner, B. The impact of fiber laser technology on the world wide material processing market. In *CLEO: Applications and Technology*; Optical Society of America: Washington, DC, USA, 2013; p. AF2J.1 .
4. Salgues, B. *Society 5.0: Industry of the Future, Technologies, Methods and Tools*; John Wiley & Sons: Hoboken, NJ, USA, 2018.
5. Boyle, A.; Meighan, O.; Walsh, G.; Mah, K.W. Laser Machining System and Method. US Patent 7,887,712, 15 February 2011.
6. Ohsawa, Y.; Kido, H.; Hayashi, T.; Liu, C.; Komoda, K. Innovators marketplace on data jackets, for valuating, sharing, and synthesizing data. In *Knowledge-Based Information Systems in Practice*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 83–97.
7. Ohsawa, Y.; Kido, H.; Hayashi, T.; Liu, C. Data jackets for synthesizing values in the market of data. *Procedia Comput. Sci.* **2013**, *22*, 709–716. [CrossRef]
8. Ohsawa, Y.; Benson, N.E.; Yachida, M. KeyGraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In Proceedings of the IEEE International Forum on Research and Technology Advances in Digital Libraries-ADL'98, Santa Barbara, CA, USA, 22–24 April 1998; pp. 12–18.
9. Aoyagi, Y.; Tani, S.; Kobayashi, Y. Pulse-by-pulse measurement of ablation volume with deep learning. *Jpn. Soc. Appl. Phys.* **2017**.
10. Kobayashi, Y.; Tani, S. Automated data acquisition and deep learning in a laser processing. In *JSAP-OSA Joint Symposia*; Optical Society of America: Washington, DC, USA, 2018; p. 19p_231B_9 .
11. Tani, S.; Aoyagi, Y.; Kobayashi, Y. Neural-network-assisted in situ processing monitoring by speckle pattern observation. *arXiv* **2020**, arXiv:2006.11351.
12. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2414–2423.
13. Monteiro, R.; Bastos-Filho, C.; Cerrada, M.; Cabrera, D.; Sánchez, R.V. Convolutional neural networks using fourier transform spectrogram to classify the severity of gear tooth breakage. In Proceedings of the 2018 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), Xi'an, China, 15–17 August 2018; pp. 490–496.
14. Mills, B.; Heath, D.J.; Grant-Jacob, J.A.; Xie, Y.; Eason, R.W. Image-based monitoring of femtosecond laser machining via a neural network. *J. Phys. Photonics* **2018**, *1*, 015008. [CrossRef]
15. Zhang, Y.; Yang, Q. A Survey on Multi-Task Learning. *arXiv* **2017**, arXiv:1707.08114.
16. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on MACHINE Learning, Helsinki, Finland, 5–9 July 2008; pp. 160–167.
17. Deng, L.; Hinton, G.; Kingsbury, B. New types of deep neural network learning for speech recognition and related applications: An overview. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–30 May 2013; pp. 8599–8603.

18. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
19. Ruder, S. An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv* **2017**, arXiv:1706.05098.
20. Sener, O.; Koltun, V. Multi-task learning as multi-objective optimization. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 527–538.
21. Evgeniou, T.; Pontil, M. Regularized multi–task learning. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 109–117.
22. Misra, I.; Shrivastava, A.; Gupta, A.; Hebert, M. Cross-stitch networks for multi-task learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3994–4003.
23. Pearson, K. On lines and planes of closest fit to systems of points in space. *Philos. Mag.* **1901**, *2*, 559–572. [CrossRef]
24. Hansen, P.C. The truncatedsvd as a method for regularization. *BIT Numer. Math.* **1987**, *27*, 534–553. [CrossRef]
25. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
27. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
28. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8026–8037.
29. Smith, L.N. Cyclical learning rates for training neural networks. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 464–472.
30. Hendrycks, D.; Mazeika, M.; Kadavath, S.; Song, D. Using self-supervised learning can improve model robustness and uncertainty. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 15637–15648.
31. Bengio, Y.; Courville, A.; Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef] [PubMed]