

Vision-Based Melt Pool Monitoring for Wire-Arc Additive Manufacturing Using Deep Learning Method

Chunyang Xia (✉ cx772@uowmail.edu.au)

Shandong University <https://orcid.org/0000-0002-3943-1321>

Zengxi Pan

University of Wollongong

Yuxing Li

University of Wollongong

Huijun Li

University of Wollongong

Research Article

Keywords: WAAM, additive manufacturing, computer vision, deep learning, anomaly monitoring

Posted Date: December 13th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-1126853/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at The International Journal of Advanced Manufacturing Technology on February 2nd, 2022. See the published version at <https://doi.org/10.1007/s00170-022-08811-2>.

Abstract

Wire-arc additive manufacturing (WAAM) technology has been widely recognized as a promising alternative for fabricating large-scale components, due to its advantages of high deposition rate and high material utilization rate. However, some anomalies may occur during the deposition process, such as humping, spattering, and robot suspend. This study proposed to apply Deep Learning in the visual monitoring to diagnose different anomalies during WAAM process. The melt pool images of different anomalies were collected for training and validation by a visual monitoring system. The classification performance of several representative CNN architectures, including ResNet, EfficientNet, VGG-16 and GoogLeNet, were investigated and compared. The classification accuracy of 97.62%, 97.45%, 97.15% and 97.25% was achieved by each model. The results proved that the CNN models are effective in classifying different types of melt pool images of WAAM. Our study is applicable beyond WAAM and should benefit other additive manufacturing or arc welding techniques.

1. Introduction

As a direct energy deposition (DED) process, wire-arc additive manufacturing (WAAM) has emerged as a suitable alternative for fabricating medium-to-large size metal components. WAAM employs an electrical arc as the heat source to fuse welding wire, and deposits layer-by-layer to form a three-dimensional object [1]. Normally, the types of electrical arc adopted in WAAM mainly include gas metal arc (GMA), gas tungsten arc (GTA), and plasma arc (PA). Compared to other forms of AM, the major advantages of WAAM are its high deposition efficiency and high material utilization (up to 100%). Normally, the deposition efficiency for laser and electron beam additive manufacturing is about 2–10 g/min, while deposition rate of WAAM ranges from 50 to 130 g/min [2]. On the other hand, WAAM possesses the advantages of lower capitalized cost, as the equipment in the WAAM system is easily available from an array of suppliers in the mature welding industry. Compared to electron beam-based AM, vacuum environments are not required for the WAAM process. In comparison to laser-based methods, the WAAM method offers a higher efficiency heat source, especially for those reflective metal alloys, such as aluminium, copper, and magnesium. Due to these advantages, WAAM has extensive application prospects in maritime [3], aerospace [4], and automotive industries [5].

To ensure the manufacturing quality, improve the automation level, and meet the industrial requirement for WAAM, it is crucial to develop reliable monitoring systems, which are capable of identifying abnormal and defects during the WAAM process. In recent years, some profound studies on monitoring and control for the WAAM process have been established. For example, Xia., Pan [6] developed a visual sensing system to monitor the geometry of melt pool during the WAAM process and implemented real-time feedback control. Chabot, Rauch [7] utilized a thermal camera to monitor the thermal distribution and history during the WAAM process. Zhan, Liang [8] applied a welding camera to monitor wire deflection during the WAAM process. Zhao, Guo [9] collect the spectrum signal and welding pool image to monitor the anomaly during WAAM. However, there still isn't any study monitoring the anomalies in melt pool. During WAAM process, some anomalies can be generated, such as hump and spatter, etc., due to the poor

process parameters or equipment setting. When those anomalies are generated, the melt pool will exhibit different morphology. Through classifying the melt pool images, it's able to detect the anomalies generated during WAAM process in real time.

During recent years, Artificial Intelligence (AI) technology has been developed remarkably and has been applied in various fields. In the laser AM field, there have been several attempts of applying AI algorithm in process monitoring and optimization. Aminzadeh and Kurfess [10] utilized a Bayesian classifier to classify the layer surface quality during powder-bed additive manufacturing. Defective and unacceptable building regions can be detected. Scime and Beuth [11] applied a convolutional neural network (CNN) to achieve autonomous powder bed anomaly detection and classification. Caggiano, Zhang [12] collected the real-time images during the Selective Laser Melting (SLM) process, and the detection of the defective condition-related pattern can be implemented through automated image feature learning and feature fusion using CNN models. Kwon, Kim [13] developed a deep neural network to classify the melt pool images with respect to 6 laser power labels in SLM. This work was expected to detect the abnormal and separate the defective parts non-destructively.

From the technological perspective, WAAM has a similar process with conventional arc welding. In welding fields, researchers have also proposed to apply machine learning technology recently. Liu, Bao [14] utilized a CNN-LSTM model to classify the images during CO₂ welding. The welding images were categorized into welding through, welding deviation, and normal welding. This work was expected to achieve online defects recognition for CO₂ welding. Xia, Pan [15] developed a vision-based monitoring system for the keyhole GTAW process, and a CNN model (ResNet) was utilized to recognize different welding states. To remove the noise in the weld pool images, Feng, Chen [16] employed a generative adversarial network (GAN) model to generate de-noised images. Furthermore, a voting-based ensemble model combining multiple CNNs was proposed to classify images from multisource sensing methods, including active vision, passive vision, and reverse electrode images (REIs). Based on this model, different welding penetration states can be identified. In research by Ren et al. (2020), the welding sound data was transformed into logarithmic time-frequency spectrograms and used as the input of the CNN model for welding penetration states classification.

To our best knowledge, few studies were conducted to expand the application of Deep Learning to WAAM fields. Therefore, this study proposed to employ Deep Learning in the process monitoring of WAAM. The deposition process of WAAM is a time-consuming process, which may cost a lot of manpower. If a monitoring system can be developed, which can detect anomaly automatically. It will help improve the production quality and save labour cost. Our study is expected to diagnose different process abnormal during the WAAM process, including humping, spattering, and robot suspend. In this study, the suitability of several state-of-the-art CNN architectures will be assessed.

The rest of this paper is organized as follows. Section 2 introduces the methodology used in this study, mainly include the CNN theory, the architecture of each CNN model, and the establishment of the dataset. Section 3 presents the result of model assessment by different criteria and discusses the results. Section

4 concludes this article. In this sense, our paper makes the following contributions: a visual sensing system for melt pool of WAAM was developed. To achieve anomaly detection, this study proposed to apply CNN in classifying different melt pool state. To our best knowledge, it's the first try to apply CNN in the process monitoring of WAAM. The state-of-art CNN architectures was investigated and compared. This proposed framework could be applied in the automation system for WAAM, which will promote the industrialization for WAAM. It is applicable beyond WAAM, and would benefit other additive manufacturing and arc welding techniques.

2. Methodology

The abnormal states investigated in this study were categorized into four types: humping defects, robot suspend, spatter, and normal state. The humping phenomenon is a common defect during the welding process, which is produced by the combined action of surface tension force and fluid flow patterns [17]. The hump can be produced when improper welding process parameters are used, such as heat input, welding speed, the flux of shielding gas, and chemical composition of the base metal. The humping defect compromises the mechanical property and geometrical integrity of the deposited component. Therefore, it is necessary to implement real-time monitoring for humping defects. During the WAAM process, due to the programming bugs, displacement restrictor of robot, improper operation, and other equipment abnormalities, the robot may be suspended during the depositing process. At the same time, the welding machine is still working. If this abnormal situation can't be diagnosed timely, the deposited part can be damaged, and even the safety problem can be caused. Spatter can be generated due to the unstable welding process, lack of shielding gas, and contamination in the material. It can be also viewed as a defect during the WAAM process. When these anomalies were produced, the melt pool will exhibit different characteristics. Through learning and classifying different melt pool images, it's able to achieve the detection of process anomalies during WAAM.

The framework of this study is as following: firstly, the dataset of different welding states was collected by experiments and labelled. And then the dataset was split into a training dataset and testing dataset. The training dataset was used for model training and the testing dataset was used for model assessment. To increase the diversity of the training dataset, data augmentation [18] was performed, including reflection, scaling, rotation, and translation. Four novel CNN architectures that differ in structure and depth were investigated, including GoogLeNet, VGG16, ResNet, and EfficientNet. To promote the training process, the models adopted pre-trained weights that learned from other datasets, such as ImageNet [19]. The block diagram in Figure 1 provides an overview of the proposed framework in the training phase.

2.1 Deep convolutional neural networks

Nowadays, deep learning has become a popular method in machine vision and pattern recognition due to its superiority in feature learning and image classification. In the fields of deep learning, CNN (convolutional neural networks) is one of the most popular algorithms, which has been successfully

applied in object detection, action recognition, and image classification [20] [21]. A classical CNN architecture consists of a series of successive layers, such as convolutional layers, pooling layers, dropout, and fully connective layers. The convolutional layer plays a most basic role, which extracts features from the image and connects nodes to a small subset of spatially connected neurons in the input image channels. The main tasks of the pooling layer are to perform downs-sampling and restore vital information in the image. It could simplify the spatial dimensions of feature maps, reduce the number of parameters, and prevent overfitting. The last layer on CNN is fully connected layer. The fully connected layer is responsible for extracting high-level features from input. In this layer, each neuron is fully connected with the previous layer, and all the inputs were combined to generate categories as output. Through combining those layers in different strategies, a series of novel architectures has been established.

2.1.1 GoogLeNet

GoogLeNet is introduced by Szegedy, Liu [22], which utilizes a CNN architecture with 22 layers. In GoogLeNet, a new conception called “Inception Module” is introduced (as shown in Figure 2), which is comprised of a shortcut branch and some deeper branches, and there are nine such modules in GoogLeNet. In inception module, a 1×1 convolution layer is applied to reduce the dimension followed by expanded 3×3 and 5×5 convolution layers. Through reducing the computation bottleneck, the depth and width of the network can be increased. In the last, the 3×3 and 5×5 convolution layers are concatenated with a parallel 1×1 convolution layer. These modules could help increase the number of units in the architecture while keeping the level of computational complexity. Instead of utilizing fully connected layers, a global average pooling is employed at the end of the network.

2.1.2 VGG

VGG networks are one of the widely used CNN models with deep architectures. The depth of this model is increased to 16 (VGG-16) and 19 (VGG-19) layers and number of parameters are reduced by using very small (3×3) convolution filters. VGG architecture has secured first and second places on the image classification and localization tasks in the ImageNet ILSVRC2014 challenge. The basic feature of this model is use of several successive convolutional layers. VGG-16 model consists of a series of successive 3×3 convolutions, 2×2 max pooling layer, and three fully connected layers with the final layer as the Softmax output. The convolutional layers are put into five groups, and adjacent groups are linked by a max pooling layer. The number of convolution filters keep constant with one group and doubles after each max-pooling layers. There are 12 convolutional layers and 3 fully connect layers in VGG-16. In this work, a VGG-16 was employed to perform classification for welding images.

2.1.3 ResNet

The ResNet model was proposed by Liu, Fang [23], which got a first in the ILSVRC 2015. When the architectures of CNN become deeper, the rapid degradation problem may be raised. To solve this problem, a novel residual blocks was proposed out. As shown in Figure 3, the shortcut connections were

introduced in the residual block, which skips multiple network layers. When the input is x , the learned feature is written as $H(x)$. The aim of residual learning is to get the residual $F(x)$ ($F(x) = H(x) - x$) between the learned features and the input. ResNet solves the problem of vanishing and exploding gradients, which exist in traditional deep neural networks. In this study, ResNet-50 was selected, which has 50 convolutional layers.

2.1.4 EfficientNet

EfficientNet was proposed to promote the performance of CNNs by scaling in three dimensions, including width, depth, and resolution, which utilizes a set of fixed scaling coefficients to meet some specific constraints. The basic block in EfficientNet is named MBConv, which is an inverted bottleneck Conv. Shortcuts are applied between bottlenecks by connecting a smaller number of channels (compared to expansion layers). It was integrated with an in-depth separable convolution, which reduced the computation by almost k^2 compared to traditional layers. Where k presents the kernel size, it specifies the height and width of the 2-dimensional convolution window. When pretrained on ImageNet and fine-tuned on Food-101, EfficientNet-B7 yields state-of-the-art top-1 accuracy of 93%, which is equal to the performance of Google's Gpipe model, with 8.7 times fewer model parameters. This is an important development—as the memory requirements for model inference become higher as model parameters increase, which means that better performance can be achieved with more modest hardware.

EfficientNet is particularly useful for using deep learning on the edge, as it reduces compute cost, battery usage, and also training and inference speeds. This kind of model efficiency ultimately enables the use of deep learning on mobile and other edge devices.

2.1.5 Transfer learning

There are mainly two methods for training CNN models: (1) Training from scratch. (2) Training using transfer learning. For CNN training, a large-scale training dataset could contribute to a generalizable result. However, in practical application, it may be not easy to obtain large-scale labelled datasets. Also, the training time for training from scratch may be too long. Additionally, overfitting is another potential risk. To promote these issues, transfer learning was proposed to retrain CNN models. In the transfer learning strategy, the weights of convolutional layers are firstly transferred from a pre-trained model to a new network that needs to be trained. In the new network, the weights of the convolutional layer are fixed, while the fully connected layers are retrained for the new task. Transfer learning is a convenient and effective method to acquire knowledge from related fields, which eliminates the need to restart the learning of the entire knowledge system.

2.2 Data collection

2.2.1 Experimental system

In this study, the dataset was collected by experiments. As shown in Figure 4, the experimental system consists of central computer, Fronius CMT welding machine, protective gas source, Xiris XVC-1000E

welding camera, and ABB robot. A 1.2 mm diameter of steel ER70S-6 filler was selected as the work material because of its extensive application in WAAM. The experiments were performed on a Q235 substrate, with the dimensions of 150×150×10 mm. The chemical composition of the used wire and substrate are presented in Table 1. A Xiris XCV-1000e welding camera was utilized to capture the melt pool image during the experiment process. To restrain the strong arc light, a filter with 650 nm central value was combined with the camera. Table 2 presents the used parameters for the welding camera.

Table 1
ER70S and Q235 Chemical compositions & ranges (wt.%).

| Alloy | C | Mn | Si | Cu | S | P | Fe |
|---------|-------|-----------|-------|------|-------|-------|-----|
| ER70S-6 | 0.08 | 1.53 | 0.88 | 0.18 | 0.01 | 0.09 | Bal |
| Q235 | ≤0.17 | 0.35-0.80 | ≤0.35 | - | ≤0.40 | ≤0.35 | Bal |

Table 2
Parameter of XIRIS welding camera

| Exposure time | AGC latency | Video average | Saturation | Gamma |
|---------------|-------------|---------------|------------|-------|
| 25ms | 75% | 4 | 40% | 50% |

2.2.2 Dataset

Table 3
Strategy of anomaly generation

| Anomaly | Strategy of generation |
|---------------|---|
| Humping | Travel speed: 500 mm/min, WFS: 5 m/min, deposition position: horizontal |
| Spattering | Shielding gas flux: 5-10 L/min |
| Robot suspend | Programming implementation |

As presented in Table 3, different strategies were applied to produce the defective samples for the dataset. For example, to collect humping samples, the welding position was set to be at horizontal position (Yuan, Pan et al. 2020). When making the spattering samples, a lower shielding gas flux was used for welding (5-10 L/min).

As shown in Figure 5, four categories of melt pool images were collected: humping, spattering, robot suspend, and normal. It can be seen that the melt pool exhibits different morphology under different welding states. In humping state, the edge of the melt pool presented large fluctuation, while the normal melt pool has a smoother edge. The spattering phenomenon can be clearly observed in the images of spattering sample. When robot suspend happened, the melt pool exhibit ellipse shape. The macro morphology of the welding sample was presented in Figure 6.

The dataset was divided into two parts randomly: the training set and the testing sets. The proportion of training and testing sets were 70% and 30%, respectively. The amount distribution of training and validation datasets for each class are presented in Table 4.

Table 4
Data for training and testing

| Category | Training set | Testing set |
|---------------|--------------|-------------|
| Robot Suspend | 2363 | 580 |
| Normal | 7469 | 2176 |
| Humping | 3876 | 964 |
| Spattering | 1590 | 397 |

2.3 Implementation

The CNN models were established by programming in Python. The machine learning library of Pytorch was utilized, and the training process was implemented on Google Colab. Google Colab is a free cloud service and supports free GPU [24]. In this study, the Numpy, Pandas, and Matplotlib libraries were also applied.

3. Results And Discussion

The learning rate is a vital parameter in the learning algorithms. There hasn't been a specific formula to calculate the learning rate. It is usually obtained by the trials and errors method. If a poor value is selected, the loss function has a chance to fall into local optimum. As a result, the network's performance will decline. In this study, different common values of learning rate were tried (as shown in Table 5). It can be seen that the best performance can be obtained when learning rate is selected as 0.005. Stochastic Gradient Descent (SGD) optimizer with momentum was utilized to minimize the loss function, since plain SGD can make erratic updates on non-smooth functions. SGD with momentum updates the weights with the moving average of the changes in individual weights for a single training sample. Table 6 presents the details of the parameters used for training. The classification accuracy of GoogLeNet, VGG-16, ResNet, and EfficientNet trained by transfer learning are all above 97% (97.25%, 97.15%, 97.62%, and 97.45% respectively), and ResNet achieves the highest accuracy among those models, which is 97.62%.

Table 5
Performance of ResNet trained by various learning rates

| Learning rate | 0.1 | 0.05 | 0.01 | 0.005 | 0.001 |
|---------------|--------|--------|--------|--------|--------|
| Accuracy | 14.09% | 14.09% | 52.85% | 97.62% | 96.43% |

Table 6
Training parameters

| Parameters | Values |
|---------------|--------|
| Max epochs | 25 |
| Batch size | 12 |
| Learning rate | 0.001 |
| Momentum | 0.9 |
| Optimizer | SGD |

A learning curve can visualize the incremental evaluation of a classifier's learning performance over time. The accuracy and loss over epochs for individual networks evaluated by the training and validation sets are illustrated in Figure 7. An epoch means one cycle of the weights updated by the full training dataset. The values of loss donate the sum of errors made by each image during the training process. A small value for loss indicates better results. It means that the classifier learned the feature of training and validation dataset with fewer errors. And a high value of accuracy indicates a better learning performance of the classifier.

The training convergence of these networks was achieved after 20 epochs, which is logical for the CNN models since they have been pre-trained by ImageNet. The fluctuations on the graph are indicative of the cost function falling into some local minima. As inferred from Figure 7, all CNN models achieved high accuracy in the training phase. The loss value of training and validation for various classifiers ensures the learning and generalization ability of these classifiers. It can be seen that the loss values for training process ranges from 0 to 0.6, and decreased rapidly with the training process.

During the training phase, it can be seen that the initial accuracy for EfficientNet is relatively low. However, it has a strong learning ability, and its accuracy increased rapidly with the training process. ResNet has a higher initial training accuracy and faster convergence rate. In the testing phase, it can be seen that the loss value and accuracy have obvious fluctuation for VGG-16 and EfficientNet, which means the training falls into local optimum, while ResNet has a smooth curve. This kind of dual learning curve helps in evaluating and selecting the suitable classifier model with an optimized loss and maximum classification accuracy.

To further investigate the classification performance of different models, the confusion matrixes were presented in Figure 8. In the confusion matrices, the row represents the actual class and the column represents the predicted class. It can be seen that ResNet could obtain the highest overall accuracy for the melt pool image classification with a value of 97.62%. For all models in this study, the highest classification accuracy was obtained for the first class: Robot Suspend, in which all the samples were categorized correctly. This is because the features of the first class are far apart from other classes. Therefore, it able to separate them in the feature space by linear SoftMax function. The least accuracy was found for the class of spattering. This result can be explained by the relative closeness of the

spattering, humping and normal class in some cases. The better performance of ResNet compared to other models may be due to its deeper architecture. ResNet-50 model has the largest deep learning structure among these CNN architectures (EfficientNet, VGG-16, and GoogLeNet). However, it should be noticed that the increase in the number of layers doesn't always ensure better performance, and overfitting may be induced. EfficientNet could achieve a balance between model accuracy and training time. Its accuracy is just a little lower than ResNet, while its training time is the two-thirds of ResNet. The lowest overall classification accuracy was achieved by VGG-16. The main reason for the low accuracy of VGG-16 compared to the other models is probably the relatively shallow architecture.

Additionally, the performance metric is derived from the confusion matrices, which could help us to better judge the performance of the classifiers. A performance metric for GoogLeNet, VGG-16, ResNet and EfficientNet is presented in Table 7. The performance metrics consist of precision, recall, and F1-score. Precision reflects how well the reference images of ground truth are classified. Precision is formed as the following equation:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

Where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively. Recall is the ratio of a correctly predicted class instance (TP) to the total number of instances associated with all classes. It shows how well a certain class has been classified [25]. The equation for Recall is presented in equation (2).

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

As shown in equation (3), when both Recall and Precision are averaged according to their weight, it becomes F1-Score.

$$F1 - score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (3)$$

From Table 7, it can be seen that all CNN models could distinguish different welding images with precision over 93%. Particularly, the ResNet model ensured the best performance for each class, which obtain the highest value in precision, recall, and F1-score. It can be also seen that VGG-16 obtained the least outcome in terms of pre, recall, and F1-score. EfficientNet could obtain a slightly lower precision than Resnet. As a general tendency, it can be observed that the performance increases with the overall complexity of the models.

Table 7
Classification values of different deep neural networks trained by transfer learning for melt pool images (%).

| Model | Index | Robot suspend | Normal | Humping | Spattering |
|--------------|----------|---------------|--------|---------|------------|
| ResNet | Pr | 99.49 | 98.95 | 98.44 | 95.65 |
| | Re | 100 | 99.31 | 97.93 | 94.21 |
| | F1-score | 99.74 | 99.13 | 98.18 | 94.92 |
| EfficientNet | Pr | 99.49 | 98.90 | 98.22 | 94.68 |
| | Re | 100 | 99.22 | 97.41 | 94.21 |
| | F1-score | 99.74 | 99.06 | 97.81 | 94.44 |
| VGG-16 | Pr | 99.49 | 98.45 | 98.52 | 93.42 |
| | Re | 100 | 99.13 | 96.89 | 92.95 |
| | F1-score | 99.74 | 98.79 | 97.70 | 93.18 |
| GoogLeNet | Pr | 99.49 | 98.45 | 97.39 | 97.13 |
| | Re | 100 | 99.22 | 96.78 | 93.70 |
| | F1-score | 99.74 | 98.83 | 97.09 | 95.38 |

As mentioned above, the deep architecture of CNN may lead to a serious complexity and computational cost. The CNN models used in this study are summarized with their properties in Table 8. As inferred from Table 8, EfficientNet-b0 has the least parameters and VGG-16 has the most parameters.

Figure 9 presents the consuming time of one epoch training for each model. It can be seen that GoogLeNet required the shortest training time for one epoch (758 s), while VGG-16 took the longest time for training (2276 s). It should be noticed that the training times of the networks depend on the hardware resources. Using advanced GPUs can ensure shorter training times for the CNNs.

Table 8
Amount of model parameters

| Network | Parameters (M) | Image input size (pixel ²) |
|----------------------|----------------|--|
| GoogLeNet [22] | 7 | 224 × 224 |
| VGG 16 [26] | 138 | 224×224 |
| Resnet 50 [27] | 25.6 | 224×224 |
| EfficientNet-b0 [28] | 5.3 | 224×224 |

To intuitively investigate the effectiveness of CNN in classify welding images, t-SNE (T-distributed Stochastic Neighbour Embedding) algorithm was employed to visualize the intrinsic feature learned by CNN models. The T-SNE algorithm has been widely recognized as an effective method to evaluate different types of features. Through located on a two-dimensional (2D) map, high-dimensional features and original samples can be intuitively visualized. Figure 10 presents the t-SNE visualization results for each CNN models in this study. Each coloured point represents a sample in the validation dataset projected from the multidimensional output of the CNN models. It can be seen that for each model, samples from the same class gather together, and each class is separable. The separability in the t-SNE algorithm investigates that all the CNN architectures used in this study are capable to classify the melt pool images of WAAM efficiently.

4. Conclusion

To diagnose the anomalies generated during WAAM process, a visual monitoring system for the melt pool was developed. In this study, the anomalies were categorized into humping, spattering, robot suspend, and normal. To recognize these anomalies automatically, convolutional neural networks (CNN) were employed. The datasets for training and validation were collected by experiments. Different novel CNN architectures were applied and compared, including GoogLeNet, VGG-16, ResNet, and EfficientNet. During the training phase, transfer learning strategy was applied. The results demonstrated that CNN could obtain high accuracy in classify melt pool images. The classification accuracy of 97.25%, 97.15%, 97.62%, and 97.45% was obtained by GoogLeNet, VGG-16, ResNet, and EfficientNet, respectively. ResNet has obtained the best performance in classifying melt pool images.

This proposed work will be helpful to improve the automation level and production quality for WAAM. In the future study, the proposed framework will be generalized by expend the image classes and dataset. At the same time, the CNN models will be optimized furtherly.

Declarations

Funding

The authors gratefully acknowledge the China Scholarship Council for financial support (NO. 201704910782)✉

Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

Author Contributions

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by [Chunyang Xia], [Zengxi Pan] and [Yuxing Li]. The first draft of the manuscript

was written by [Chunyang Xia] and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript

References

- [1] Suryakumar, S., K. Karunakaran, U. Chandrasekhar, and M. Somashekara (2013). *A study of the mechanical properties of objects built through weld-deposition*. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 227(8):1138-1147. <https://doi.org/10.1177/0954405413482122>.
- [2] Suryakumar, S., K. Karunakaran, A. Bernard, et al. (2011). *Weld bead modeling and process optimization in hybrid layered manufacturing*. Computer-Aided Design, 43(4):331-344. <https://doi.org/10.1016/j.cad.2011.01.006>.
- [3] Ya, W. and K. Hamilton. *On-demand spare parts for the marine industry with directed energy deposition: propeller use case*. in *International Conference on Additive Manufacturing in Products and Applications*. 2017. Springer, p 70-81.
- [4] Williams, S.W., F. Martina, A.C. Addison, et al. (2016). *Wire+ arc additive manufacturing*. Materials Science and Technology, 32(7):641-647.
- [5] Greer, C., A. Nycz, M. Noakes, et al. (2019). *Introduction to the design rules for metal big area additive manufacturing*. Additive manufacturing, 27:159-166. <https://doi.org/10.1016/j.addma.2019.02.016>.
- [6] Xia., C., Z. Pan, S. Zhang, et al. (2020). *Model predictive control of layer width in wire arc additive manufacturing*. Journal of Manufacturing Processes, 58:179-186. <https://doi.org/10.1016/j.jmapro.2020.07.060>.
- [7] Chabot, A., M. Rauch, and J.-Y. Hascoët (2019). *Towards a multi-sensor monitoring methodology for AM metallic processes*. Welding in the World:1-11.
- [8] Zhan, Q., Y. Liang, J. Ding, and S. Williams (2017). *A wire deflection detection method based on image processing in wire+ arc additive manufacturing*. The International Journal of Advanced Manufacturing Technology, 89(1-4):755-763.
- [9] Zhao, Z., Y. Guo, L. Bai, et al. (2019). *Quality monitoring in wire-arc additive manufacturing based on cooperative awareness of spectrum and vision*. Optik, 181:351-360.
- [10] Aminzadeh, M. and T.R. Kurfess (2019). *Online quality inspection using Bayesian classification in powder-bed additive manufacturing from high-resolution visual camera images*. Journal of Intelligent Manufacturing, 30(6):2505-2523. <https://doi.org/10.1007/s10845-018-1412-0>.
- [11] Scime, L. and J. Beuth (2018). *A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process*. Additive

Manufacturing, 24:273-286. <https://doi.org/10.1016/j.addma.2018.09.034>.

[12] Caggiano, A., J. Zhang, V. Alfieri, et al. (2019). *Machine learning-based image processing for on-line defect recognition in additive manufacturing*. CIRP Annals, 68(1):451-454. <https://doi.org/10.1016/j.cirp.2019.03.021>.

[13] Kwon, O., H.G. Kim, M.J. Ham, et al. (2020). *A deep neural network for classification of melt-pool images in metal additive manufacturing*. Journal of Intelligent Manufacturing, 31(2):375-386. <https://doi.org/10.1007/s10845-018-1451-6>.

[14] Liu, T., J. Bao, J. Wang, and Y. Zhang (2018). *A hybrid CNN–LSTM algorithm for online defect recognition of CO2 welding*. Sensors, 18(12):4369. <https://doi.org/10.3390/s18124369>.

[15] Xia, C., Z. Pan, Z. Fei, et al. (2020). *Vision based defects detection for Keyhole TIG welding using deep learning with visual explanation*. Journal of Manufacturing Processes, 56:845-855.

[16] Feng, Y., Z. Chen, D. Wang, et al. (2019). *DeepWelding: A Deep Learning Enhanced Approach to GTAW Using Multisource Sensing Images*. IEEE Transactions on Industrial Informatics, 16(1):465-474.

[17] Bradstreet, B. (1968). *Effect of surface tension and metal flow on weld bead formation*. Welding journal, 47(7):314s-322s.

[18] Perez, L. and J. Wang (2017). *The effectiveness of data augmentation in image classification using deep learning*. arXiv preprint arXiv:1712.04621.

[19] Deng, J., W. Dong, R. Socher, et al. *Imagenet: A large-scale hierarchical image database*. in *2009 IEEE conference on computer vision and pattern recognition*. 2009. Ieee, p 248-255.

[20] Ren, S., K. He, R. Girshick, and J. Sun. *Faster r-cnn: Towards real-time object detection with region proposal networks*. in *Advances in neural information processing systems*. 2015. p 91-99.

[21] Schmidhuber, J. (2015). *Deep learning in neural networks: An overview*. Neural networks, 61:85-117. <https://doi.org/10.1016/j.neunet.2014.09.003>.

[22] Szegedy, C., W. Liu, Y. Jia, et al. *Going deeper with convolutions*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015. p 1-9.

[23] Liu, Z., Y. Fang, S. Cui, et al. (2016). *Stable keyhole welding process with K-TIG*. Journal of Materials Processing Technology, 238:65-72.

[24] Carneiro, T., R.V.M. Da Nóbrega, T. Nepomuceno, et al. (2018). *Performance analysis of google colab as a tool for accelerating deep learning applications*. IEEE Access, 6:61677-61685.

[25] Congalton, R.G. (1991). *A review of assessing the accuracy of classifications of remotely sensed data*. Remote sensing of environment, 37(1):35-46. <https://doi.org/10.1046/j.1365-2249.2000.01137.x>.

[26] Simonyan, K. and A. Zisserman (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556.

[27] He, K., X. Zhang, S. Ren, and J. Sun. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. p 770-778.

[28] Tan, M. and Q.V. Le (2019). *Efficientnet: Rethinking model scaling for convolutional neural networks*. arXiv preprint arXiv:1905.11946.

Figures

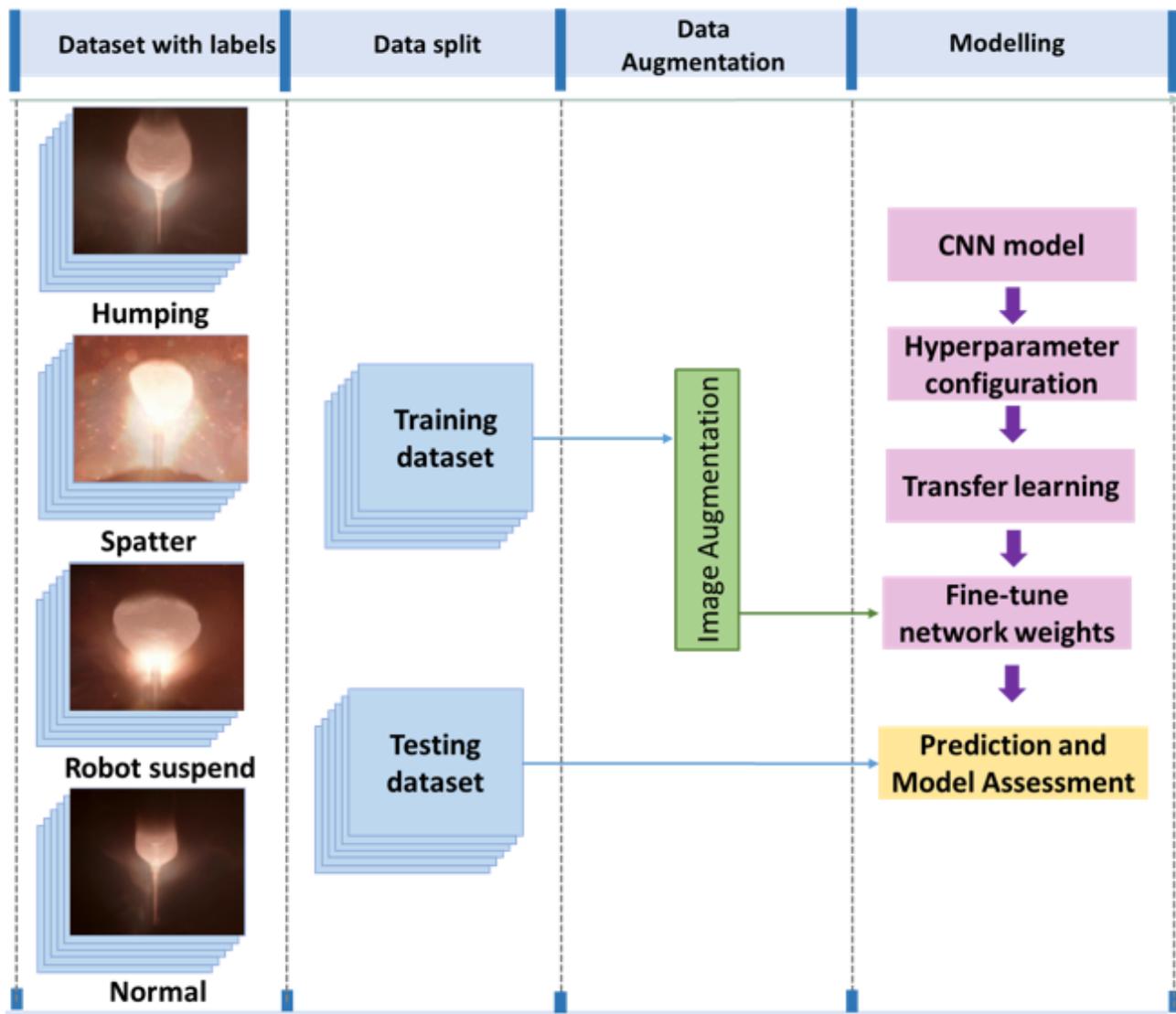


Figure 1

Overall workflow of the proposed anomalies classification framework

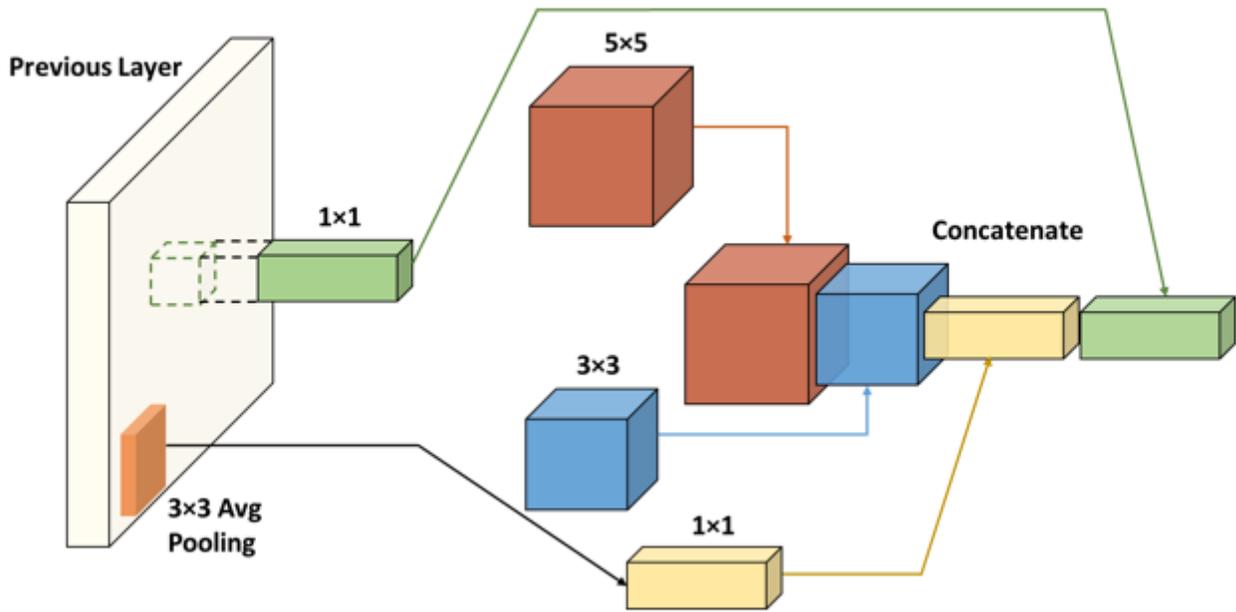


Figure 2

Inception Module

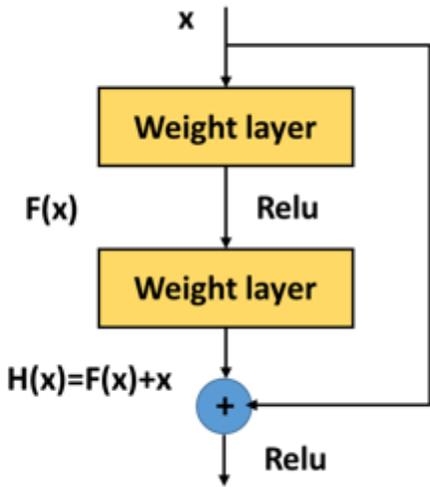


Figure 3

Single Residual block

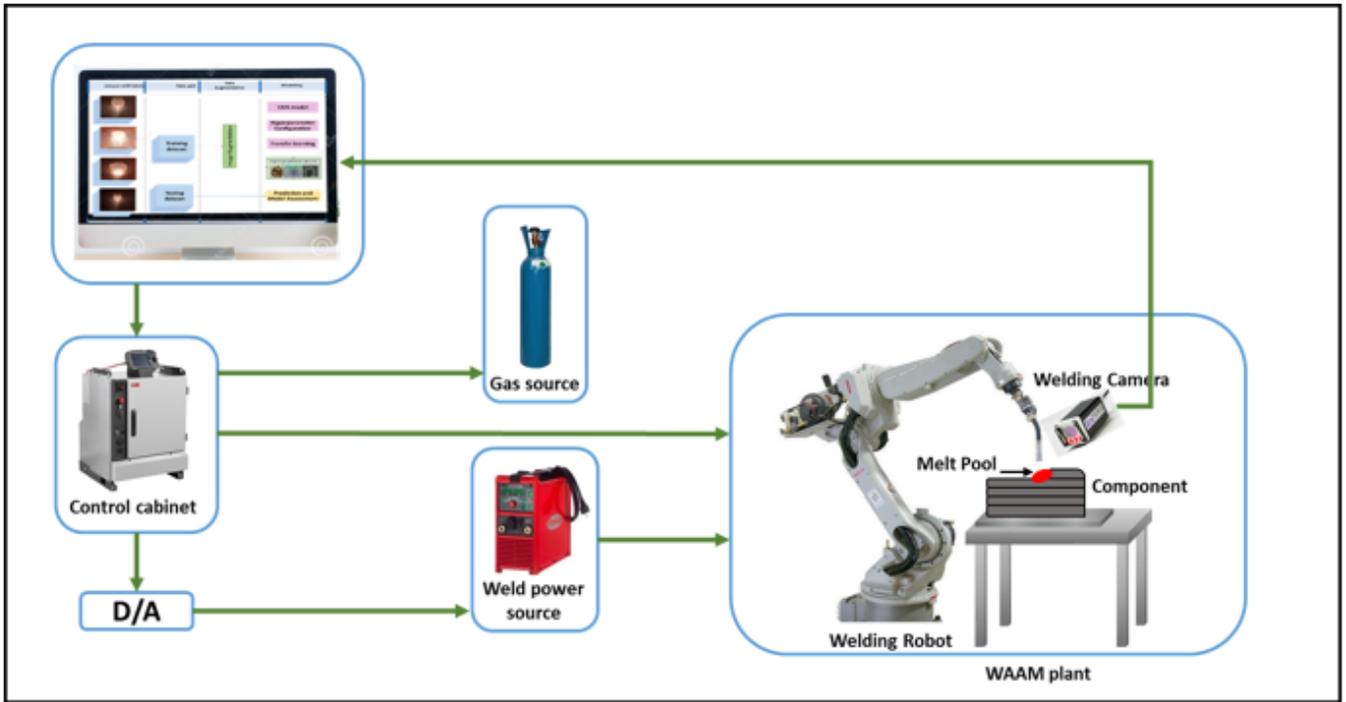
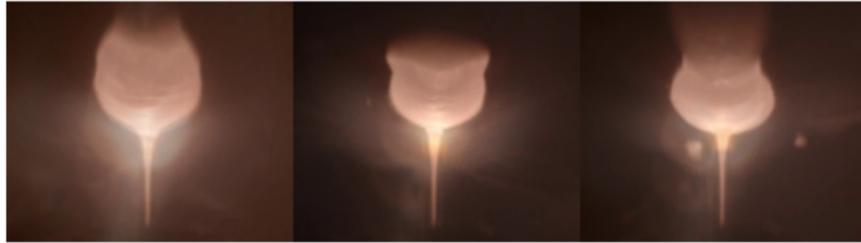


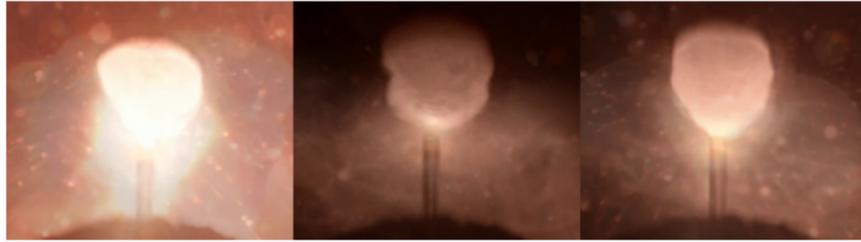
Figure 4

Experimental system

Humping



Spattering



Robot suspend



Normal



Figure 5

Sample images of the dataset

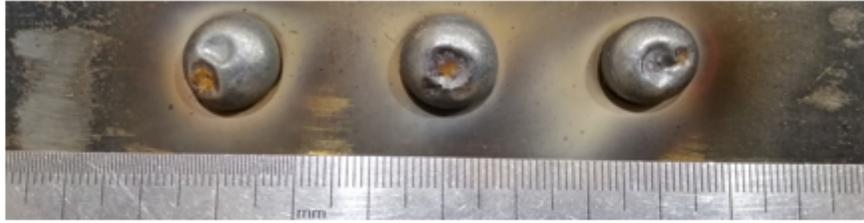
Humping



Spattering



Robot suspend



Normal



Figure 6

Specimen images

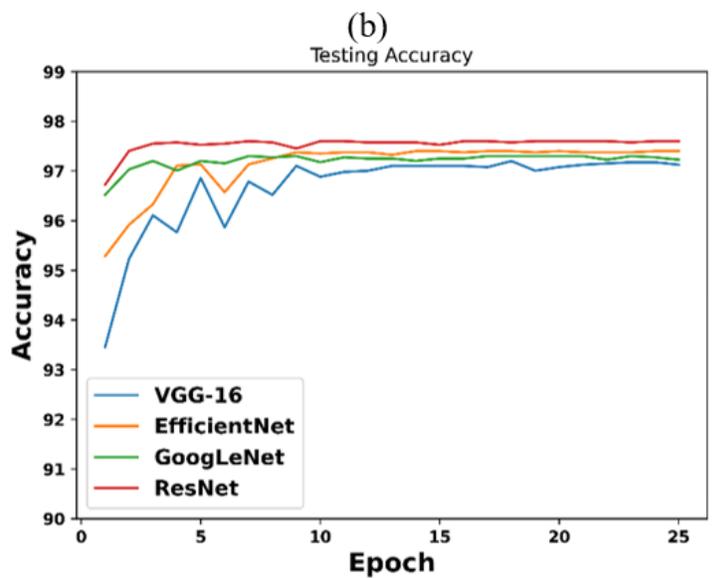
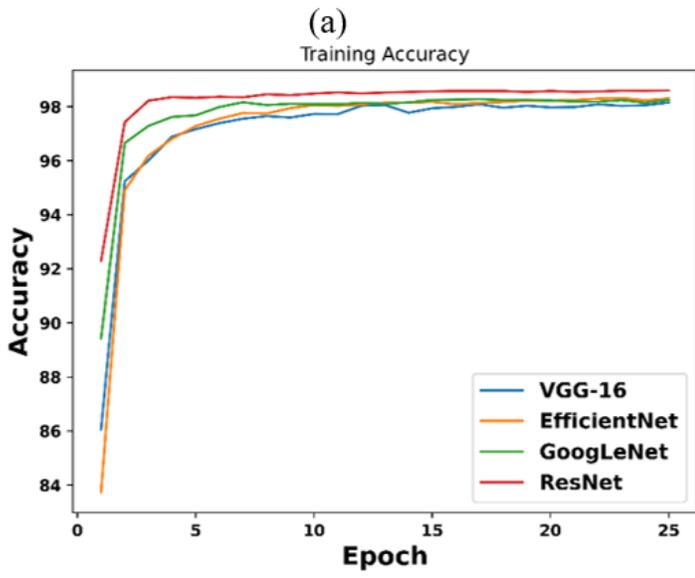
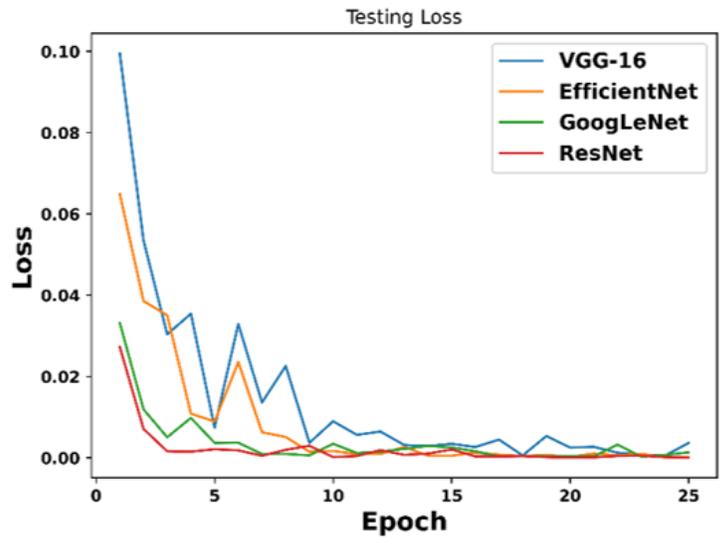
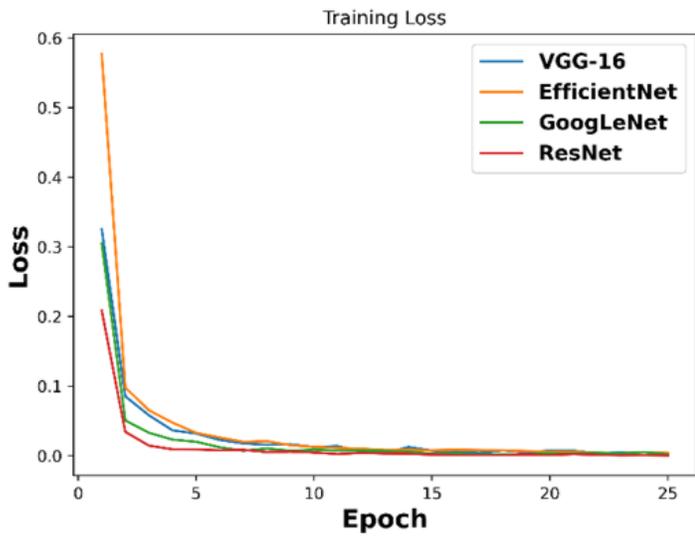


Figure 7

Training curve. (a) Training loss (b) Testing loss (c) Training accuracy (d) Testing accuracy

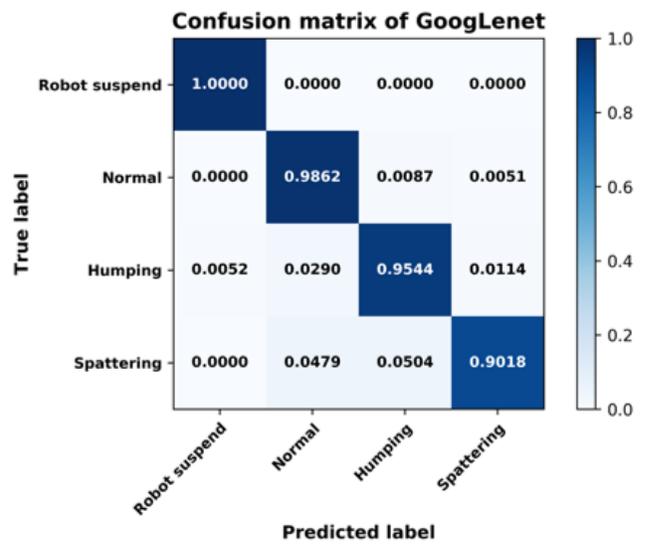
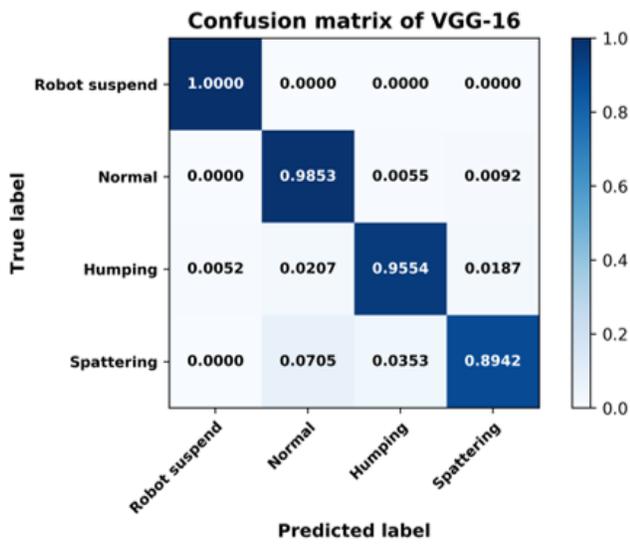
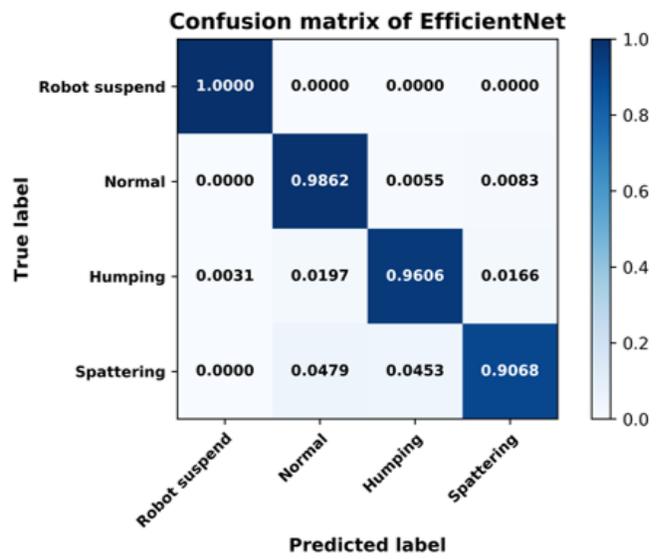
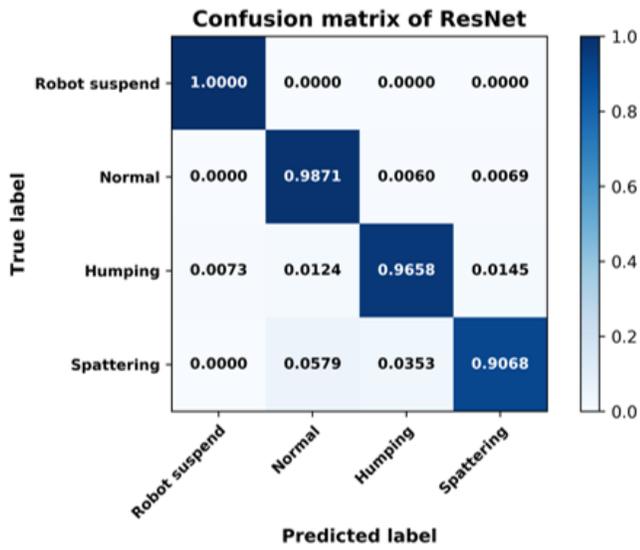


Figure 8

Confusion matrix

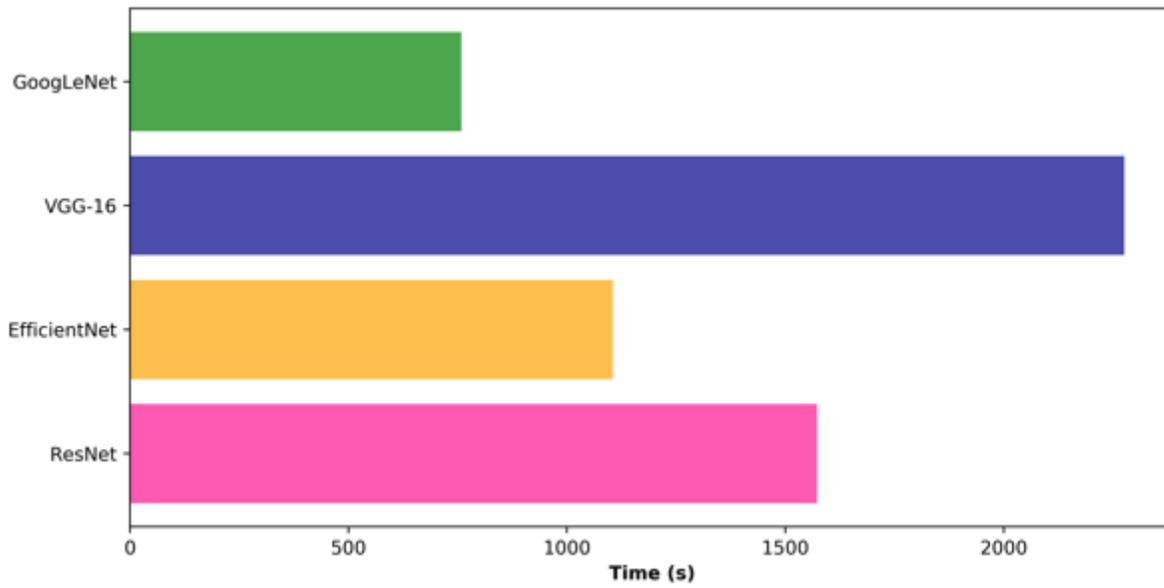
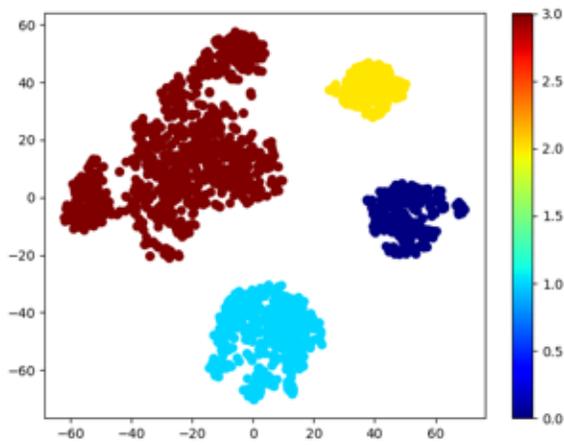
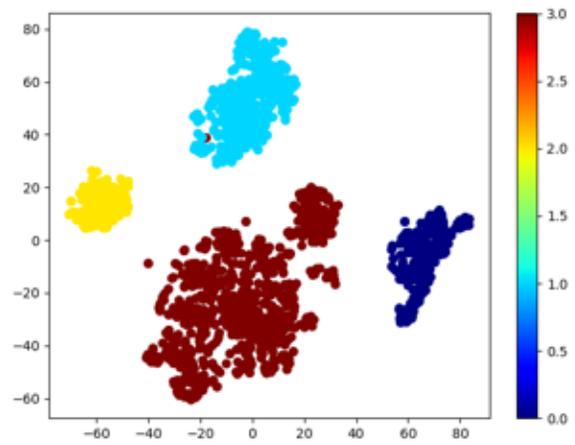


Figure 9

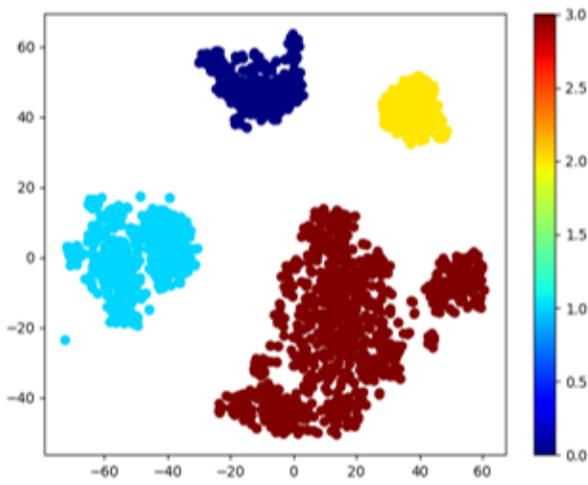
Training time



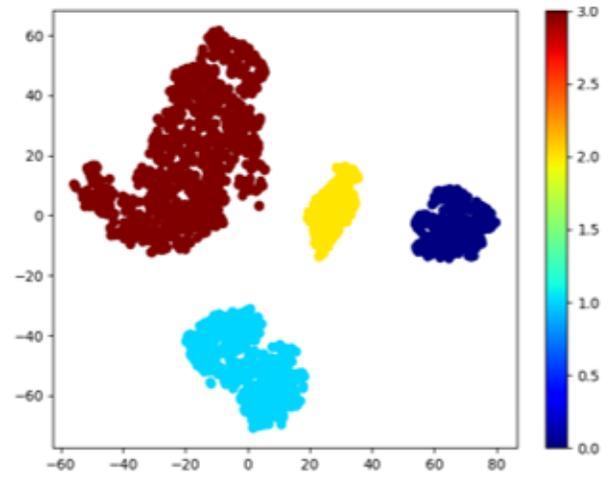
(a) ResNet



(b) EfficientNet



(c) GoogLeNet



(d) VGG-16

Figure 10

T-SNE visualization for CNN. (a) ResNet. (b) EfficientNet. (c) GoogLeNet. (d) VGG-16