# Instance segmentation of on-line wear debris using deep convolutional neural network with transfer learning

## Introduction:

One important trend for mechanical equipment’s maintenance is the application of condition monitoring techniques. The idea is to analyze real-time data to determine the machine’s health and respond only when maintenance is actually necessary [[1, 2](#ref1)]. Currently, techniques that get focused on include vibration analysis, acoustic emission with ultrasound, oil analysis and wear debris analysis (WDA). Each one has its own advantages and constraints. It’s widely accepted that a single technique cannot meet the requirements for all situations, still, wear debris analysis is considered as one of the most effective approach by many users of condition based maintenance techniques for the following reasons:

* Wear debris or wear particles are generated as components move relative to each other. The parameters that define wear debris such as their shape, size, surface texture, etc. reflect the wear modes, wear mechanisms, and the severity associated with their generation [[3](#ref1)[, 4](#ref10)].
* The patterns of quantity and type of wear debris changes over time have strong correlation with state of machine. It is possible to predict potential failures or component’s deterioration from very early stage to avoid catastrophic accidents [[5, 6](#ref10)].

Although WDA can provide a lot of information for problem detection and fault diagnosis, the technique has not been widely used in the industry for that the debris’ morphology assessment, classification and the machine status evaluation relies heavily on expertise, which is time-consuming, costly and objective. These limitations have driven the need to automate this technique from which the industry will benefit in terms of economy and safety [[7, 8](#ref10)].

The availability of modern instrumentation provided a wide array of studies that helped identify the characteristics of wear debris and the mechanisms by which they generated [[9](#ref10)]. Approaches by observing and analyzing the morphological features of wear debris are thus promising potential online solutions [[10, 11](#ref10)]. Until now, researchers have made considerable efforts to build auto-classification system for wear debris. It is reported that various machine learning algorithm can be used to distinguish adhesive, fatigue and abrasive particles using area, perimeter and elongation parameters [[12, 13, 14](#ref15)]. These studies showed that these simple parameters are effective for certain types of wear debris. However, there are several kinds of abnormal debris larger than 20 cannot be accurately identified. Wu Tonghai team carried out extensive research on the on-line monitoring of wear debris [[15, 1](#ref20)6, 17]. The team had independently developed a set of online visualized ferrography and proposed that the area of debris coverage and their diameters in the visible area can be used as indicators for judging the state of machine. This study provided a hardware design for on-line ferrography. However, the detailed surface texture of the debris was not obtained due to small magnification.

Another problem drawing a lot of attention is the auto-segmentation of wear debris’ image. As a crucial first step for auto-classification, the accuracy of segmentation has great impact on the subsequent processes. The complicated background of the image and debris’ overlapping phenomenon makes segmentation a hard job. It’s reported in several literatures that ant-colony algorithm, gray level integrated morphological features and J-segmentation algorithm can be used to solve this problem [[7, 1](#ref20)8, 19]. The drawback of these methods is that they all require carefully pre-processing and parameter configuration, the result of segmentation will change dramatically under different situations.

To sum up, a lot of research had been done on on-line wear debris’ analysis. The recognition of cutting, spherical and oxide abrasive grains has basically been realized through shape, size and color while classification of similar debris such as severe sliding particle, fatigue laminar particle, and chunky spalling particle is not entirely successful yet. Segmentation of overlapping debris or ones that formed in chains remains unsolved.

Deep neural networks (DNNs), also referred to as deep learning, are a sub-field of machine learning which aims at giving computers the ability to learn without being explicitly programmed [20]. DNNs are currently widely used for many artificial intelligence jobs including computer vision, speech recognition, and robotics [[21, 22](#ref20)]. DNNs have broad prospects for wear debris’ detection, yet relevant research is nearly blank so far.

The objective of this study is to solve the problems described above. A deep convolutional neural network is used for wear debris’ segmentation and classification at the same time, also known as instance segmentation which combines object detection and semantic segmentation. This study focused on practical matters such as building dataset consist of various kinds of wear debris, architecture setting and testing of transfer learning procedure. State-of-art accuracy of identifying as well as localizing debris from images of large magnification is achieved.

## 2. Methodology

**2.1 Data collection**

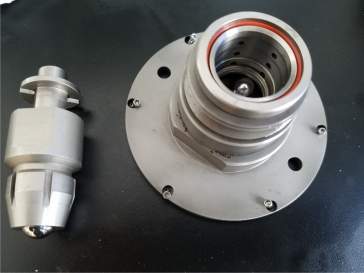
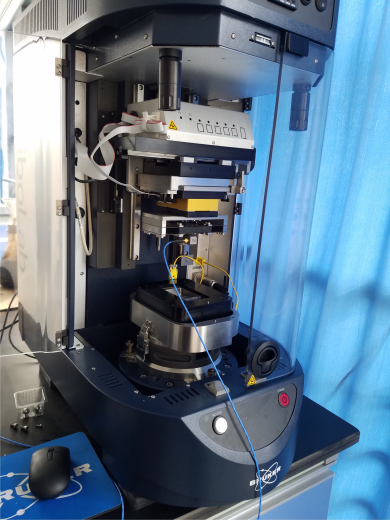
***2.1.1 Generation of wear debris***

The testing machine, showed in fig.[1](#fig2), has a universal base that can be equipped with a range of drive modules simulating rotational, linear, or oscillating motions as well as an upper carriage fitted with a force measuring sensor. Three kinds of tribology tests were performed to simulate different wear modes:

**Pin on disk** consists of a stationary pin loaded against a rotating disk. (The tests were conducted in a laboratory air with temperature of 22℃ and a relative humidity of 50%). The upper pin made of standard 416 stainless steel is cylindrical in shape with polished flat ends of 8*mm*, the disk is made of alloy steel E52100 with surface roughness RA=0.02μm. The Mobile Gard 412 was used as lubricant. The tests were conducted at 900r/m, which means the linear velocity reached 170mm/s. A load of 30kg was applied to the pin for 25h. Pin on disk test was mainly for generating adhesive wear debris. Adhesive wear particles were the main product of this test. Adhesive wear is the transfer of material from one contacting surface to another. The tests were set this long so that both mild and severe debris can be generated. Some pictures of adhesive debris are showed in first row of fig2. They are mostly small particles formed in chains, especially in last two pictures debris stick together due to magnetization which is quite common in ferrograph [[7](#ref20)].

Compared to pin on disk test, **Pin on plate test** replaced the rotating module by a reciprocating module. The upper pin is made of cast iron HT250 and the plate is made of Gr15 steel with surface roughness RA=0.2μm. The stroke was set to 18mm and frequency was set to 4Hz. The wear particles were generated with a 48kg load for 12h. Unlike pin on disk test, pin on plate test was mean to simulate the wear condition of reciprocating friction pairs such as piston-ring cylinder or hydraulic cylinder. It can be seen from the pictures in second row of fig.[2](#fig4), some of the severe sliding wear particles are quite large with clear surface scratch due to rough surface and long hours of wear process.

The fatigue wear particles were generated by **4-ball testing** machine. The material of the ball is GCr15 (hardness 63*HRC*). Maximum load and speed were set to 900N and 300r/min. In order to generate fatigue wear particles the running time was set to 50h. Fatigue begins with reduced lubrication regime and continuous stresses that exceeded the endurance limit of the material, causing cracks beneath the surface. This creates micro-pitting and eventually destructive spalling. The fatigue particles are demonstrated in the third row of fig.[2](#fig4), there are some micro spalling particles in the first two pictures and some large chunky spalling and laminar particles in last two pictures.



(a)Bruker UMT’s base (b) four-ball module and pin on disk module

Figure 1 Bruker’s UMT universal mechanical tester.

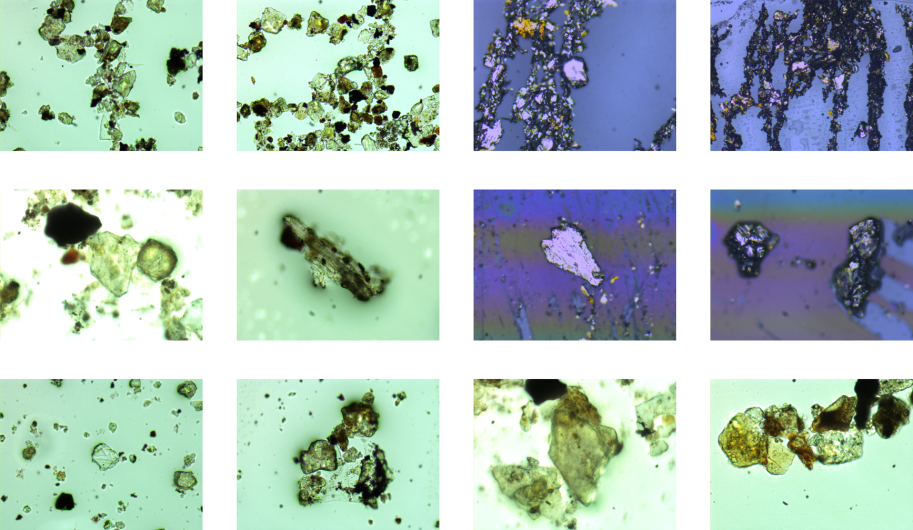


Figure 2 Examples of wear debris generated in the tests.

***2.1.2 Semi-on-line sampling***

The most unreliable part of wear debris analysis is the method used to extract wear debris from the system. In order to take microscopic pictures from a flowing fluid, the objective lens must be very close to the debris which is exactly the cause for many problems (larger the magnification, shallower the depth of focus). For example, a window or a transparent section must be designed otherwise nothing in the tube can be seen. The difficulty is that large magnification needs both inner diameter and thickness of that section to be minimized. The section would be so fragile that flow rate would be nearly zero if it won’t explode at all. As a consequence, researchers only uses 20× magnification, and even with that, restoration techniques has to be applied to enhance the quality of the part that out of focus [15].

In this study, a semi-on-line approach by Ferrography is proposed. Ferrography uses a high-gradient magnetic field to extract wear debris from a fluid sample as it flows down a specially prepared microscope substrate. Since the debris can be attached to the surface, it’s possible to alter the window section with a detachable bottom to separate the work into two steps: debris collection and image capture. One can control the time intervals of these two steps depend on the machines he/she wants to monitor. The distribution and quantity of debris overtime could still provide valuable information.

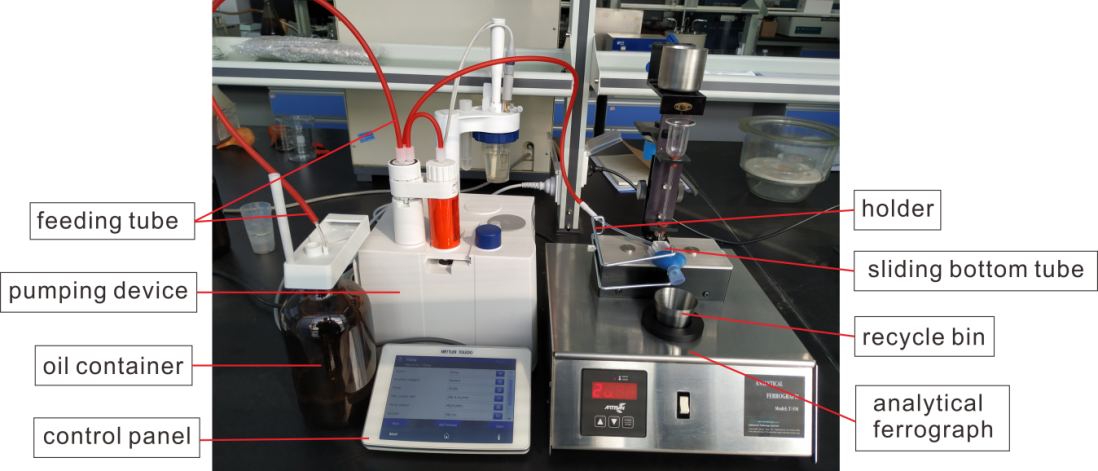


Figure 3 Connection of Analytical ferrogram and lubricating oil cycling system.

.

To simulate such semi-on-line system, lubricating oil was put in a container connecting to analytical ferrogram. The oil flow is driven by a small pump. It goes out of the container and through the tubes, finally down to an acrylic section with sliding bottom.as showed in fig.[3](#fig3). The sliding bottom has a double concave rail structure while the middle part is flat so it can hold the debris and be put under the lens. During the process, the rails are sealed with a thin layer of bending PVC as shown in fig.4. Since the Bruker’s UMT can’t attach to the system directly, lubricating oil were extracted after each kind of test and put into the container. The container, pump and tubes served as a flow system of on-line sampling process.

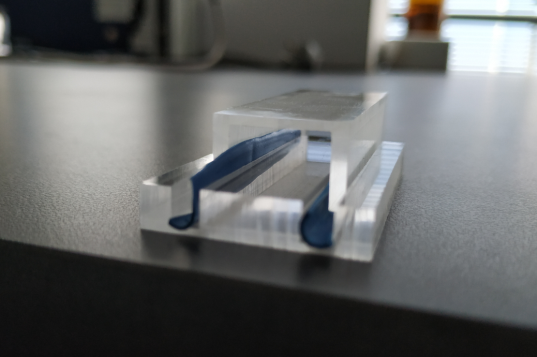
******

Figure 4 Sliding bottom with simple sealing

In this experiment, images were taken every 30 minutes of collecting. In the images capture step, the pump is stopped and the bottom detached, the residual fluid on it is washed off (another advantage of the design) before it be put under the microscope. Images were shot by optical microscopy OlypusBX51 with a Color Charge Couple Device (CCD) Camera. Most of the pictures were taken under 200× magnification because it produces sharp and clear image of debris of size 20μm to 200μm which are of particular interest in this study.

**2.2 Deep neural network**

***2.2.1 Pipe line of instance segmentation***

In this research, we focused on instance segmentation of **five kinds of wear debris** [[2,](#ref10)[1](#ref30)3]:

* Normal rubbing wear: small platelets particles less than 15μm in major dimensions.
* Cutting wear debris: curls or spirals like particles generated by abrasive wear, can be 25-100μm long and 5um thin.
* Severe sliding debris: large flaks with striation on the surface, indicating local adhesion, more than 20um in major dimension.
* Laminar rolling fatigue debris: larger than 20um in major dimension with relatively smooth surface.
* Chunky fatigue spall: chunks of rough metal, irregular shape, indicating severe fatigue spalling.

As briefly mentioned above, the instance segmentation is challenging because it requires the correct detection of all wear debris in an image while precisely segmenting each one. In this experiment, the Mask-RCNN was used to train on the dataset of wear debris described in the previous section [23]. Mask-RCNN consists of two parts: A region proposal network (RPN) to generate a set of dense candidate regions from a set of anchors containing a foreground object, i.e. wear debris; A classification network for debris’ classification, bounding box regression and prediction of segmentation mask accordingly. The main advantage of using a deep neural network is that it can learn to classify and detect object from real data. Providing better accuracy and robustness than tradition hand engineered classifier with features such as color, shape or texture property.

***2.2.2 Network Architecture***

Convolutional Neural Networks is a common form of DNNs for computer vision tasks [[24](#ref15)]. Krizhevsky et al. [[22](#ref15)] used the CNN with extended depth to obtain the best classification accuracy at the ImageNet Large Scale Visual Recognition Challenge (LSVRC) in 2012. Since then, CNN models have been dominated in all kinds of image recognition competitions. The local connections, weight sharing and pooling operations of CNNs effectively reduce the complexity of the network and therefore reduce the number of training parameters. Also the model has some degree of invariance to translation, distortion, and scaling. The networks are composed of multiple convolutional (CONV) layers as shown in fig.[5](#fig6). In such networks, each layer is supposed to generate a successively higher-level abstraction of the input data, called a feature map, which holds important unique information. Nowadays, a deep neural network can contains a thousand CONV layers [[25](#ref20)], giving the model extremely representation power. Fully-connected layers (FC) serve the purpose of final classification in the end of structure. The CONV/FC can be calculated as follows given the shape parameters in Table 1:



**O**, **I**, **W** and **B** are the matrices of the output feature maps, input feature maps, filters and biases, respectively. *U* is a given stride size. Non-linearity is referred to a non-linear activation function. Typically it is applied right after CONV or FC layer, the non-linearity is necessary for the model since the network can be simplify to a single linear function mathematically without it. The most popular non-linearity used is rectified linear unit (ReLU) because of its simplicity [[26](#ref30)]. From the demonstration figure one can see that there are also pooling and normalization layers. The definition and effect of these layers is introduced in [[27](#ref30), 34].

Table.1 Shape parameter of CONV and FC

|  |  |
| --- | --- |
| Shape Parameter | Description |
| H/W | Input feature map’s height/width |
| R/S | Filter’s height/width ( = H or W in FC) |
| E/F | Output feature map’s height/width ( = 1 in FC) |
| N | Batch size of 3-D feature maps |
| C | Number of filter/input feature map channels |
| M | Number of total filters/ Number of output feature map channels |

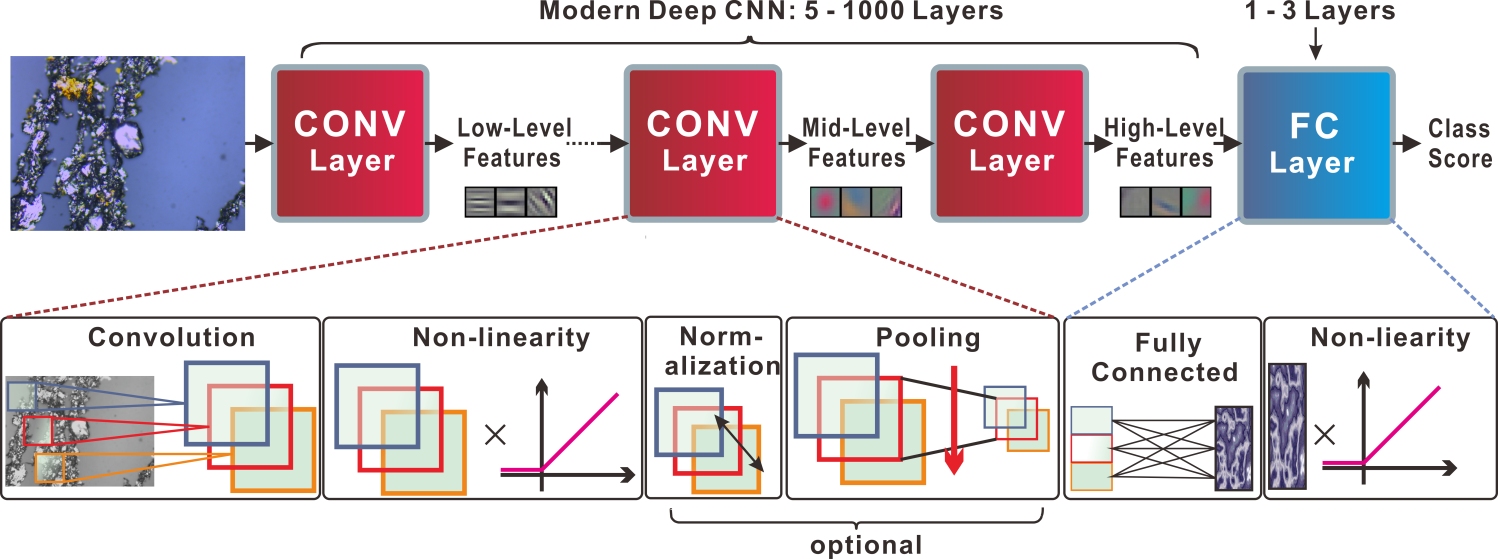
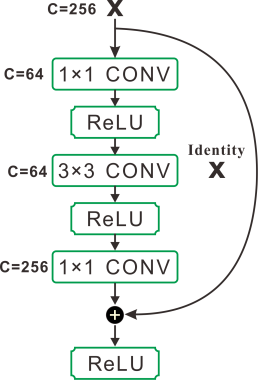
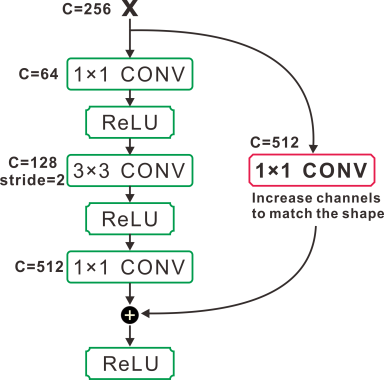


Figure 5 Framework of typical modern deep convolutional neural network.

ResNet-101 was adopted in this experiment in which each stage is composed of a number of residual blocks. As demonstrated in fig.[6](#fig9), instead of learning the function for the weight layers *F(x)*, residual block provides a shortcut that learns the residual function *F(x) = H(x)-x*). The residual connection is shown effective in very deep neural networks. The detailed layer structure of ResNet-101 is showed in table [2](#tab2). One thing to note is that in Conv phase2, 3 and 4, the first block increases dimensions as the one showed in fig.[6(b)](#fig9). The first 3×3 CONV layer and residual double the number of channels while stride is set to 2.

(a) Typical Residual block (b) Residual block that increases dimension

Figure 6 A diagram of Residual connection

Table2: Architectures of 3 CNN models. The convolutional layer parameters are denoted as ‘Receptive field, Number of channels’. The ReLU activation function and BN is not shown for brevity.

|  |  |  |
| --- | --- | --- |
| Net | Output size | Layer detail |
| Conv phase 0 | 112×112 | Conv7×7,64, stride 2 |
| Conv phase 1 | 56×56 | Maxpool3×3, stride 2 |
| Conv phase 2 | 28×28 |  |
| Conv phase 3 | 14×14 |  |
| Conv phase 4 | 7×7 |  |

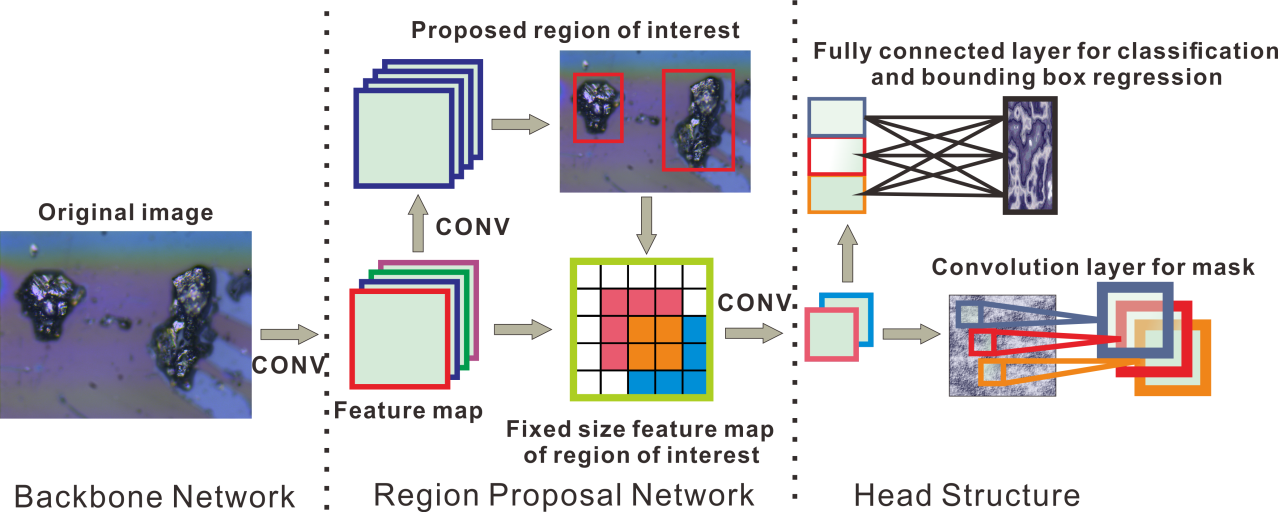
******

Figure 7 A diagram of Mask-RCNN structure

The ResNet-101’s phase one to three is the backbone feature extractor of the whole network while the phase four is the feature extractor of the final classification head. The data then flow through a FC layer for classification and bounding box regression, in parallel, the data flow through a small fully convolutional network (FCN) to generate the final mask for segmentation [28]. The RoIAlign layer is used to transform the feature map of every RoI proposed by the RPN to a fixed size feature map. Then the fixed sized feature map is fed into the head network for classification, bounding box regression and mask generation. The detailed structure is well described in [23, 29].

## 3. Transfer learning implementation

Training an entire Mask-RCNN from scratch is impractical, because it is relatively rare to have a dataset of sufficient size. What's commonly done is using a CNN which have been trained on a huge dataset and make use of the information in there, i.e., transfer learning. Study shows that it’s almost always better to use transfer learning on new dataset even when the new dataset is very different from original dataset [[30](#ref33)].

In this experiment, the Mask-RCNN pre-trained with COCO dataset was used as benchmark which contains more than 200 thousands labelled images with 80 categories of object [31]. RPN was trained with labelled wear debris’ images to enhance the RPN’s ability to identify background and wear debris. The RPN was trained end-to-end by stochastic gradient descent (SGD) with back-propagation [32]. Each mini-batch arises from a single image that contains 64 positive and negative anchors as described in [23]. The object lost function over a mini-batch is composed of classification loss and regression loss:

Here, *i* is the index of an anchor in a mini-batch, and are predicted probability and ground-truth probability of anchor *i*. Similarly, and represent the 4 parameterized coordinates of the predicted bounding box and ground-truth box associated with positive anchor.

, ,

, ,

, ,

, ,

Where *x*, *y*, *w* and *h* denote the box’s center coordinates and its width and height. The classification loss is the cross-entropy function while the regression loss is the smooth L1 function across all anchors:

In the following, we train Mask-RCNN’s head using 2,000 RPN proposals each image. Transfer learning of the head of Mask-RCNN requires some alternation of structure. The last FC layer of classification branch is replaced with a soft-max classifier for 6 classes since five different types of wear debris in our dataset, not 80 classes in COCO dataset, the extra one class is for background. The final layer of regression branch and mask branch are altered accordingly too. For regression branch, the original FC layer with size of 81 is replaced with a FC layer with size of 6. For mask branch, the last CONV layer is replace with CONV layer with 6 filters instead of 81. The loss function of head classification network is similar to the RPN Loss. The key difference between the RPN layer and the classification layer is that while the RPN dealt with just two classes: foreground and background, the head deals with all the object classes (plus background). The bounding box regression loss is calculated similar to the RPN except the network calculates regression coefficients for each object class. The mask branch has 6 14×14 dimensional output for each region proposed by RPN, which encodes 6 binary masks of resolution 14×14, one for each of the classes. is calculated using the average binary cross-entropy loss:

Where is the sigmoid function:

The other implement details are as follows: Each mini-batch contains 1 image and each image has 64 sampled RoIs (16 of foreground and 48 of background). Learning rate was set to 10-5. Input images are resized such that their scale (shorter edge) is 800 pixels and mean subtraction was used on the training set. We use a weight decay of 0.0001 and a momentum of 0.9. The models were built by Pytorch framework [[33](#ref33)] and run on 2 Nvidia GeForce 1080 Titan GPU for 50 iterations.

## 4. Result and discussion

In order to evaluate the effectiveness of the model, the dataset is split into training set and test set. The test set is used for evaluation of a model and never be touched in training. In total, there were 1500 images in training set and 300 images in test set. At test time, different numbers of proposals are generated by RPN. Box prediction branch is applied on these proposals. For mask branch, only the boxes within top 100 highest scores are used. The floating-number output of mask branch is then resized to the RoI size and round to 0 or 1 at a threshold of 0.5. The main criteria used in object detection and instance segmentation is average precision (AP). The AP summarizes the shape of the precision/recall curve, and is defined as the mean precision at a set of recall levels [31]. Higher AP means the localization of object is relatively better and misclassification is less. To be considered a correct detection, the mask IoU overlap must be above a certain threshold, in this experiment, 0.5. The IoU is the intersection over Union, it measure how close a given mask is to another mask calculated as the area of overlap over area of union.

First we experiment on the effect of RPN by comparing the result of original RPN and the trained RPN with different numbers of proposal. Since the benchmark Mask-RCNN was trained on the COCO dataset. The original RPN should be able to find anchors with foreground objects. The number of proposal boxes matter too, if too many proposals are send to the head structure for training, the number of ‘bad’ proposals will increase too and therefore lower the final precision, if too little, the network would be under fitting. The AP of all kinds of wear debris are shown in table3. From the table, it can be seen that trained RPN performs better than original RPN, especially on abrasive debris. The reason might be that in the original COCO dataset, there aren’t so many thin and curvy objects like abrasive debris. During training on wear debris’ images, it gradually learned how to identify these shapes. The number of boxes seems have minor influence to the final result, 300 would be appropriate choice for it boost the AP a little.

Table3: The result of RPN testing with different number of proposal boxes.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Method | # proposals | AP | Rubbing | Severe | Fatigue | Abrasive | laminar |
| original RPN | 100 | 71.9 | 76.4 | 79.6 | 72.5 | 62.5 | 68.4 |
| - | 300 | 72.4 | 78.6 | 80.1 | 73.9 | 62.4 | 67.0 |
| - | 500 | 72.2 | 78.5 | 79.4 | 72.1 | 62.9 | 68.1 |
| Trained RPN | 100 | 79.3 | 83.3 | 81.4 | 76.3 | 80.5 | 75.0 |
| - | 300 | 79.8 | 83.6 | 81.5 | 76.3 | 80.9 | 76.6 |
| - | 500 | 79.4 | 83.2 | 81.2 | 76.4 | 80.4 | 75.8 |

After evaluate RPN, two transfer learning scenarios were tested on the head of Mask-RCNN

1. **CNN as a feature extractor**. During training, freeze all the learnable parameters except the final classifier. This is based on the belief that pre-trained network could extract useful features in the image, one only need to train the final classifier. It’s not only faster but also can prevent over fitting
2. **Fine-tune Conv phase 3 and 4**. Keep earlier layers fixed during training but adjust the weights in Conv phase3 and 4 along with the classifier. This is motivated by observation that the features learned by low-level blocks are more general, less abstract than those found higher-up. Since dataset of wear debris is very different from COCO, the weight in the high-level blocks should be adjusted.

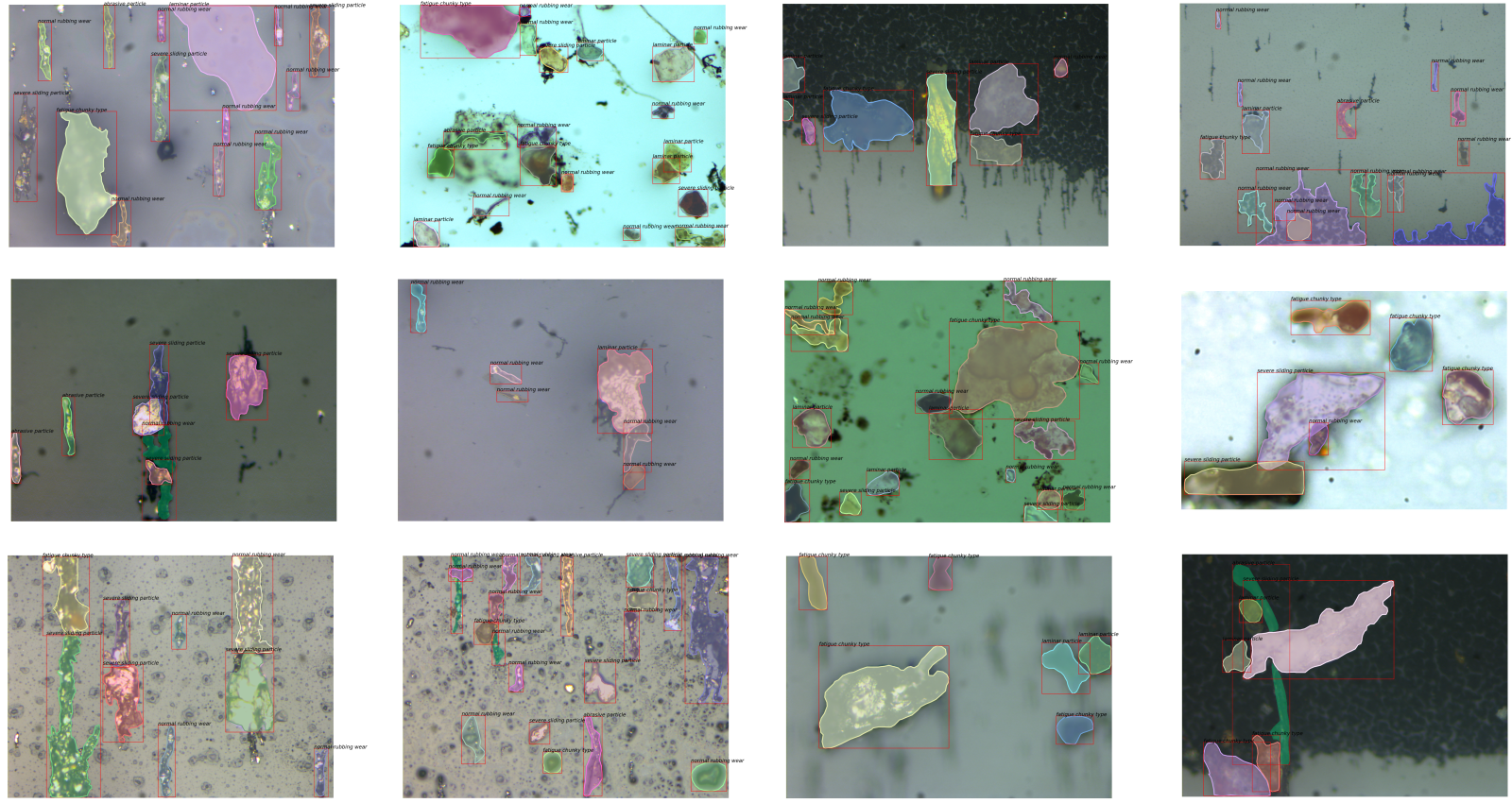
The final result is reported in Table4. In the experiment, we also calculated the AP of bounding box as shown in the last two rows of Table4. The AP of bounding box is similar to AP of mask except the IoU is based on the area of overlap and area of union of bounding box. From the final result, we can see that AP of bounding boxes in general is better than AP of masks. This is predictable because it’s harder to predict masks. More importantly, fine-tuning the Conv phase3 and 4 results in a great gain in both mask AP and bounding box AP. This suggests that the higher levels do contain abstract information about objects in COCO datasets. The model generalizes better when weights and bias in high level can be adjusted. Some outputs are visualized in Fig.8. Mask RCNN performs well even dealing with overlapping wear debris. One of the advantages of this model is that it calculates each kind of wear debris on separate mask branch. Therefore, the whole image segmentation job is disentangled. Since the model predict masks, the subsequent calculation can be done relatively easy, such as the measurement of size or area. The most common error is that it confused the shadows in the background or residual oil with debris. Very small debris cannot be detected for two reasons: Firstly, they are not labelled well in the dataset; secondly, in the pre-trained model even the smallest detecting anchor is much bigger than those isolated micro particles. Above all, mask-RCNN with transfer learning is promising algorithm for on-line WDA. 

Figure 7 Instance segmentation results on the test-set

Table4: The final result using two transfer learning strategies

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method | AP | Rubbing | Severe | Fatigue | Abrasive | laminar |
| Feature extractor | 79.8 | 83.6 | 81.5 | 76.3 | 80.9 | 76.6 |
| Fine-tune | 85.4 | 86.1 | 83.8 | 83.5 | 86.6 | 86.0 |
|  | APbb | - | - | - | - | - |
| Feature extractor | 81.5 | 85.4 | 83.6 | 77.8 | 81.3 | 79.4 |
| Fine-tune | 88.2 | 88.7 | 85.2 | 89.4 | 87.6 | 90.1 |

## Conclusion

Through a series of experiment, a datasets is built which can be valuable for future related research on wear debris analysis. A deep neural network structure suitable for real-time instance segmentation of wear debris was established. A conclusion can be drawn that deep convolution neural network is suited for wear debris’ auto classification and detection. Using transfer learning and proper training configuration, mask-RCNN achieves state of art accuracy. In further studies, a dataset of more detailed wear debris should be built, e.g. spheres, contaminants and combined rolling sliding debris in gear system. Also small-sized proposals should be focused on to improve the model’s ability to detect small debris in the images.

## References:

1. S.K. Yang, A condition-based failure-prediction and processing-scheme for preventive maintenance, IEEE Transactions on Reliability 52 (3) (2003) 373–383. doi:10.1109/TR.2003.816402.
2. S. Raadnui, Wear particle analysis utilization of quantitative computer image analysis: A review, Tribology International 38 (10) (2005) 871–878. doi:10.1016/j.triboint.2005.03.013.
3. W. Li, H. Pham, An inspection-maintenance model for systems with multiple competing processes, IEEE Transactions on Reliability 54 (2) (2005) 318–327. doi:10.1109/TR.2005.847264.
4. Kowandy, C. Richard, M. Chen, Correlation between the tribological behaviour and wear particle morphology case of grey cast iron 250 versus graphite and ptfe, Wear 262 (7-8) (2007) 996–1006. doi:10.1016/j.wear.2006.10.015.
5. R. Upadhyay, Microscopic technique to determine various wear modes of used engine oil, Journal of Microscopy and Ultrastructure 1 (3) (2013) 111 – 114. doi:https://doi.org/10.1016/j.jmau.2013.09.002.
6. S. Ebersbach, Z. Peng, N. Kessissoglou, The investigation of the condition and faults of a spur gearbox using vibration and Wear 260 (1) (2006) 16 – 24. doi:https://doi.org/10.1016/j.wear.2004.12.028.
7. J. Wang, L. Zhang, The segmentation of wear particles in ferrograph images based on an improved ant colony algorithm, Wear 311 (1-2) (2014) 123–129. doi:10.1016/j.wear.2014.01.004.
8. H. Liu, H.Wei, L.Wei, An experiment on wear particles texture analysis and identification by using deterministic Industrial Lubrication and Tribology 67 (6) (2015) 582–593. doi:10.1108/ILT-01-2015-0008.
9. J. Roylance, Ferrography then and now, Tribology International 38 (10) (2005) 857–862.
10. Sperring, T.P.; Nowell, T.J. "SYCLOPS—a qualitative debris classification system developed for RAF early failure detection centres." Tribology International 38 (2005): 898-903.
11. Bartz, W.J. Batchelor, A.W. (2000). Engineering Tribology 2nd edition. Elsevier Butterworth-Heinemann.
12. A.K. Jardine, D. Lin, D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance Mechanical Systems and Signal Processing20 (7) (2006) 1483 – 1510.
13. N. M. M. Manoj Kumar, Parboti Shankar Mukherjee, Advancement and current status of wear debris analysis for machine condition monitoring: a review, Industrial lubrication and tribology 65 (1) (2013) 3–11.
14. W. Yuan, K. Chin, M. Hua, Shape classification of wear particles by image boundary analysis using mac Mechanical Systems and Signal Processing 72 (Supplement C) (2016) 346 – 358.
15. W. Xi, T. Wu, K. Yan, Restoration of online video ferrography images for out-of-focus degradations, [EURASIP Journal on Image and Video Processing](https://jivp-eurasipjournals.springeropen.com/) (2018) 2018:31 doi: [https://doi.org/10.1186/s13640-018-0270-1 20182018:31](https://doi.org/10.1186/s13640-018-0270-1%2020182018:31)
16. H Wu, T Wu, Y Peng, Z Peng, Watershed-based morphological separation of wear debris chains for on-line ferrograph analysis. Tribol. Lett. 53(2) (2014) 411–420
17. T Wu, Y Peng, S Wang, F Chen, N Kwok, Z Peng, Morphological feature extraction based on multiview images for wear debris analysis in on-line fluid monitoring. Tribol. Trans. 60(3) (2017), 408–418
18. T.Wu, H.Wu, Y. Du, N. Kwok, Z. Peng, Imaged wear debris separation for on-line monitoring using gray level and integrated Wear 316 (1) (2014) 19 – 29. doi:https://doi.org/10.1016/j.wear.2014.04.014.
19. H. Liu, H. Wei, H. Xie, (2018) "Unsupervised segmentation of wear particle’s image using local texture feature", Industrial Lubrication and Tribology, https://doi.org/10.1108/ILT-09-2017-0275
20. Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521(7553) (2015) 436–444.
21. L. Deng, J. Li, J.T. Huang, M. Seltzer, Recent advances in deep learning for speech research at Microsoft. In ICASSP, 2013.
22. A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in NIPS, 2012.
23. K. He, G. Gkioxari, P. Doll´ar, and R. Girshick. Mask RCNN. In ICCV, 2017
24. K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.
25. K. He, X. Zhang, S. Ren, Deep Residual Learning for Image Recognition. In CVPR, 2016.
26. K. He, X. Zhang, S. Ren, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In ICCV, 2015.
27. Sze, Vivienne, et al. "Efficient Processing of Deep Neural Networks: A Tutorial and Survey." Proceedings of the IEEE 105.12(2017):2295-2329.
28. Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation In CVPR, 2017.
29. S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.
30. [J. Yosinski](https://arxiv.org/search/cs?searchtype=author&query=Yosinski%2C+J), [J. Clune](https://arxiv.org/search/cs?searchtype=author&query=Clune%2C+J), [Y. Bengio](https://arxiv.org/search/cs?searchtype=author&query=Bengio%2C+Y), How transferable are features in deep neural networks? 2014, [arXiv:1411.1792v1](https://arxiv.org/abs/1411.1792v1)
31. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll′ar, and C. L. Zitnick. Microsoft COCO: Common objects in context.
32. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard,W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1989.
33. Pytorch: An open source deep learning platform that provides a seamless path from research prototyping to production deployment. <https://pytorch.org/>
34. S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.