International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137

Volume – 07, Issue – 05, May 2022, PP – 49-72

SCAN Based Clustering of Wear Particles In Denoised-Background Removed OLVF Images Obtained Using Zoomed-Fired U-Net

Abhijit Bhandari[1],Wei Cao[2], Zili Jin[3], Jiajun Wu[4], Zhiyang Cao[5], Jianying Yan[6], Jianfeng Wu[7]

(School of Mechatronic Engineering, Xi'an Technological University, Xi'an ,710021, P.R. China)

Abstract: Different Ferrography approaches for analysing the wear particles have been devised in order to understand the Performance, Prediction of Failures and the remaining useful life of any mechanical system. Online Visual Ferrography (OLVF) is one of the best methods allowing researchers to capture images of wear particles even when the system is running. Butthe images obtained from it are low in resolution with random noise. To enhance the image properties and suppress the noise, different researchers have developed various processes. Many of the papers do not employ neural networks to solve the issue. So, to add to the solution, we have devised a U-Net architecture, i.e. the Zoomed-Fired U-Net (ZF U-Net). The ZF U-Net is a comparatively fast and robust denoising and background removal network capable of producing good results in just few milliseconds. Along with the Denoising Network, Semantic Clustering by Adopting Nearest Neighbours (SCAN) has been used to cluster the wear particles into individual clusters based on their extracted features. We have developed a pipeline that can denoise, remove the background and group the wear particles present in an OLVF Image into their respective clusters thus extending our abilities to understand more about the wear process.

Keywords: Architecture, Clustering, Denoise, OLVF, ZF U-Net

1. INTRODUCTION

We often observe that machine faults are developed first through a gradual performance deterioration stage and then speed up to a catastrophic stage. During the deterioration stage, machine components wear out and it produces wear particles. Hence, it is possible to infer the machine health state from inspecting features of wear particles [1] [2]. This idea led to the development of ferrography.

In recent scenarios, analytical ferrography is proved one of the best methods for wear characterization [3]. Analytical Ferrography despite being a highly reliable method of wear characterization, its inability to serve in real time condition monitoring led to the development of Online Visual Ferrograph (OLVF).

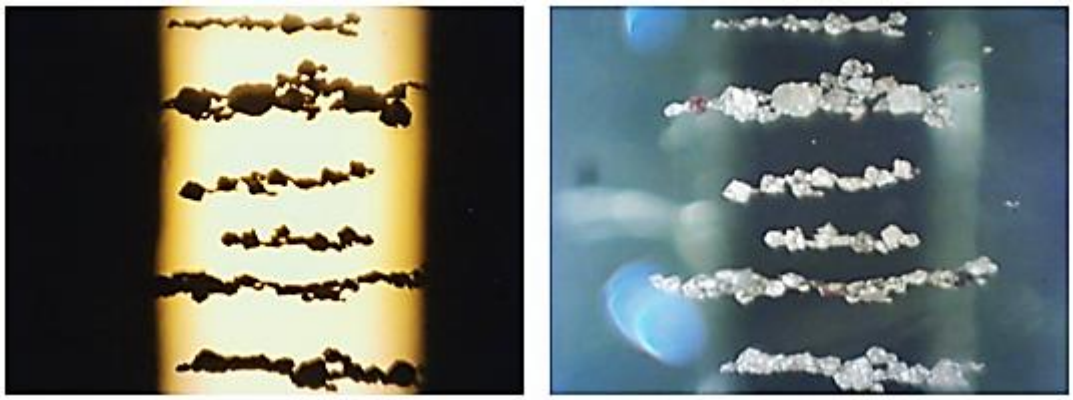
The OLVF uses a magnifying microscope and a camera to capture online videos of wear particles flowing in a short transparent length of the lubrication circuit. The captured video contains a large amount of wear particle information. The color of the particles helps to identify the degree of oxidation [3] [4]. Using online visual ferrography has some practical difficulties. We need signal and image processing routines to enhance wear particle images.

OLVF provides fuzzy information related to the wear debris, which needs a lot of tedious post - processing. It is difficult and time-consuming to identify each type of wear particles from their chains. Also, the unevenness of the light because of the varying transparency and astigmatism of the passing oil complicates background color. However, the advantages outweigh the disadvantage. The major advantage, as mentioned earlier is that, it allows real time and intensive sampling, making the entire process a true representative of the real-time wear conditions.

Based on the position of the source of light, an OLVF generated image can be of two types- a) Transmitted Image, b) Reflected Image. The contour of wear particles can be identified from Transmitted Image and Color details from the reflected image [3]. Here, in the research we have focused in enhancement and information extraction from Transmitted Image.

Many researchers have been working hard on several mathematical models and processes to extract the most well-defined contour out of the OLVF image after enhancement of the image quality.

To add on to the list of the available methods, we have developed a highly flexible, re-trainable improved version of U-Net Architecture called ZF U-Net. Fig. 1 shows the types of images that can be generated from an OLVF.



1. two types of online visual ferrograph images- (a) transmitted image (b) reflected image [source:[3]]

Adding on to the development of denoising and the background removing ZF U- Net, we have also focused on unsupervised clustering of wear particles present in every image collected into their individual clusters thus providing an ability to classify wear particles based on their properties.

In the domain of wear particle classification, it is a very challenging job to develop a model which can classify the wear particles. For the development of such a model, the most promising method is the use of Supervised Learning. Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately [5].

Here, as we don't have any annotated data or data with labels, the use of supervised learning is impossible. Hence, the only possibility is a) Use of Unsupervised Learning b) Use of Semi- Supervised Learning. Here, in this work, we focused on the use of a semi-supervised learning algorithm to classify the extracted wear particles into their respective clusters.

Thus, the main area of focus of the paper is the development of a denoising and background removal neural network architecture followed by intermediate processes and finally semi-supervised clustering of wear particles present in individual images.

2. Dataset Preparation

For supervised learning of a Deep Neural Network, it is important to have the input data and the expected output. Any supervised Deep Neural Network learns by comparing the loss between output from the network and the expected output. For our case, the noisy wear particle images obtained from the OLVF are the input sources.

For the training of our network, we have to find a mathematical model that works best in denoising of the OLVF image. Keeping in mind about random noises, we decided that underwater image processing techniques could be handy. Underwater images processing techniques have been widely used in marine energy exploration, marine environment protection, marine military and other fields [6]. We have made an extensive analysis over individual underwater image processing techniques and even their combinations to have the best denoised output to train our ZF U-Net Model. Fig. 2 shows an image captured underwater and an image obtained from the OLVF.

2. noisy and enhanced underwater image [source:[7]]

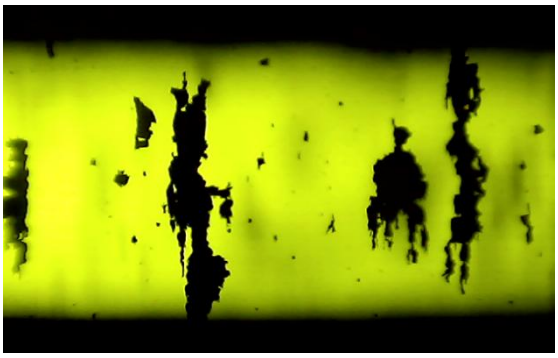


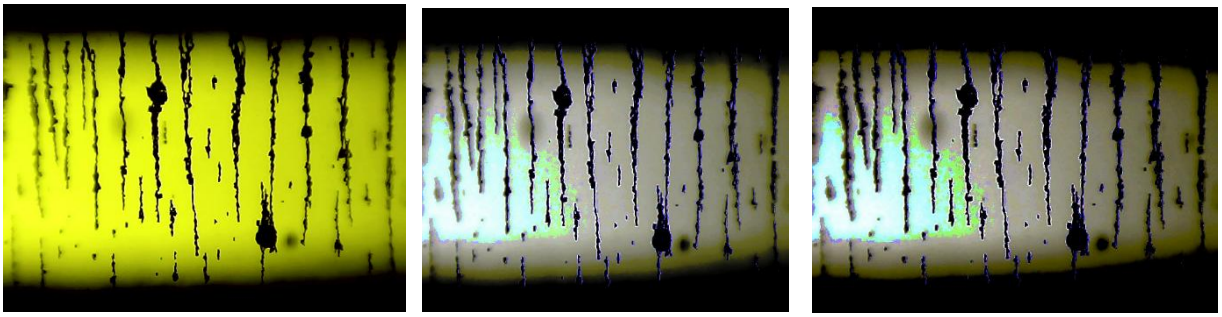
image from olvf

2.1 Underwater Image Processing

The state of art for processing underwater images is developing day by day. Whether it be for the study of marine life or military purposes, underwater image processing techniques have started producing much finer results. In order to deal with underwater image processing, we have to consider first of all the basic physics of the light propagation in the water medium. Physical properties of the medium cause degradation effects not present in normal images taken in air. Underwater images are essentially characterized by their poor visibility because light is exponentially attenuated as it travels in the water and the scenes result poorly contrasted and hazy [8].

Similar to underwater images, the images captured from OLVF suffer the attenuation when the light travels from air to the tube then to the lubricant oil. Due to multiple refractions and the vibration from the engine or the mechanical system, different random noises are encountered. Hence, considering the similarities between the source of noise, we thus concluded to review some of the best performing state of art models to generate expected output for training our network.

After extensive research and visual analysis, we came to the conclusion that feeding output from Rayleigh Distribution Model to Image Blurriness and Light Absorption Model (IBLA) helps to enhance image properties at well-lighted regions and also helps to improve the quality at blurred areas. Fig.3 shows image obtained from an OLVF, output from Rayleigh Model and output from IBLA after applying on the output from Rayleigh Model.



3. original image from olvfoutput from rayleigh modelibla applied on output from rayleigh

3.1.1 Image Enhancement Using Rayleigh Stretching and Averaging of Image Planes

Transmitted image from OLVF usually consists of bright and dark areas. Overall image contrast can be increased by the global stretching of the image. The contrast of the image needs to be increased to increase the brightness of darker areas. However, global stretching also leads to an increase in the brightness of the bright areas, leading to the over-enhancement of the bright areas. These over-enhanced areas cause image pixels to become too bright, resulting in loss of details. The same problem occurs when global stretching is applied to the darker areas because it produces under-enhanced areas that reduce image details.

Keeping that in mind, we decided to use the Rayleigh Stretching and Averaging of Image Planes which uses two different image contrasts. The idea of the method is to produce two images with different contrasts: one as an under-enhanced image and another as an over-enhanced image. These two images are produced from a single image [9].

The mid-point of the original histogram is determined to produce the under- and over-enhanced images. The histogram is then divided into two regions and stretched based on this mid-point. The stretching process is applied to the images with respect to the Rayleigh distribution [10] [11].

The two produced images are stacked together, and the average value between the two images is calculated. The image is then converted into the Hue-Saturation-Value (HSV) color model. The S and V components of the HSV color model are stretched within the dynamic range of 1% from the minimum to the maximum values. The final image is obtained by converting the image back into the RGB color model [9].

After the decomposition of image into its color channels in the RGB color model, the original histograms of the channel are divided at the mid-point of their intensity level. (1) is applied to calculate the mid-point of the intensity level, i_{mid}, of the original histogram. This equation is applied to all three channels of the RGB color model to determine the mid-point of the intensity level:

$$i_{mid} = \frac{I_{\max} - I_{\min}}{2} + I_{\min}$$

where, I_{max} and I_{min} represent the minimum and maximum intensity levels of individual channels, respectively.

After calculating the mid-point, two regions of intensity levels are produced. The lower region consists of the value between the minimum intensity level and the mid-point of the original histogram, whereas the upper region consists of the value between the mid-point and the maximum intensity level value of the original histogram. These two regions are stretched over the entire dynamic range of the output intensity level.

The original equation of histogram stretching is given by (2) [12]:

$$P_{out} = (P_{in} - i_{\min})\left(\frac{o_{\max} - o_{\min}}{i_{\max} - i_{\min}}\right) + o_{\min}$$

Where P_{in} and P_{out} are the input and output pixels, respectively, and i_{min} ,i_{max} , o_{min} and o_{max} are the minimum and maximum intensity levels for input and output images, respectively.

The probability distribution function of the Rayleigh distribution is given by (3) [12]:

$$PDF_{Rayleigh} = \left(\frac{x}{\alpha^2}\right) e^{\left(\frac{-x^2}{2\alpha^2}\right)} \quad \text{for} \quad x \geq 0, \alpha > 0$$

Where α is the distribution parameter of the Rayleigh distribution and x is the input data, which is the intensity value in this case.

(2) is integrated into (3) to apply the Rayleigh-stretched distribution, thus producing the Rayleigh-stretched distribution as in (4) [9]:

$$Rayl.-stretched = \frac{\left[(P_{in} - i_{\min})\left(\frac{o_{\max} - o_{\min}}{i_{\max} - i_{\min}}\right) + o_{\min}\right]}{\alpha^2} \cdot e^{\frac{\left[(P_{in} - i_{\min})\left(\frac{o_{\max} - o_{\min}}{i_{\max} - i_{\min}}\right) + o_{\min}\right]^2}{2\alpha^2}}$$

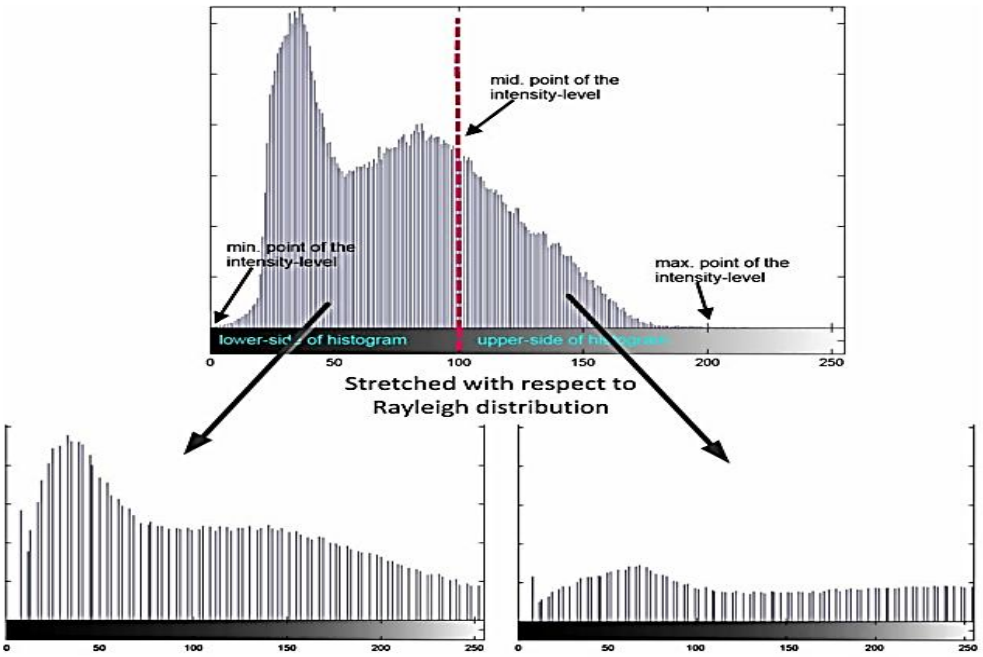
The output histogram is stretched over the entire dynamic range so the values of o_{max} and o_{min} can be substituted with the values of 255 and 0, respectively. Therefore, (4) can be simplified as (5) [9]:

$$Rayl.-stretched = \frac{255(P_{in} - i_{\min})}{\alpha^2 (i_{\max} - i_{\min})} \cdot e^{\frac{-[255(P_{in} - i_{\min})]^2}{2\sigma^2 (i_{\max} - i_{\min})^2}}$$

Where i_{min} and i_{max} indicate the minimum and maximum intensity level values for input image in each region, respectively.

For the lower region, i_{min} indicates the minimum intensity level value in each channel, and i_{max} indicates the maximum intensity level value in the region, which is equivalent to the mid-point value of the original histogram. For the upper region, i_{min} indicates that the minimum intensity level value in the region is equivalent to the mid-point value of the histogram, and i_{max} indicates that the maximum intensity level value in the region is equivalent to the maximum intensity level of the original histogram. The described processes are applied to all channels of RGB color model.

Fig. 4 shows an example of the histogram with its maximum, minimum, and mid-points. The original histogram is divided at its mid-point to produce two separate histograms. These two histograms are stretched over the entire dynamic range of [0,255] with respect to the Rayleigh distribution. After applying these processes to the corresponding channels, all lower-stretched histograms are composed to produce an image and all upper-stretched histograms are also composed to produce another image. This process produces two different images with different contrasts. The lower region of the histogram produces an under-enhanced image, whereas the upper region produces an over-enhanced image. These two images are composed using the average values between these images.



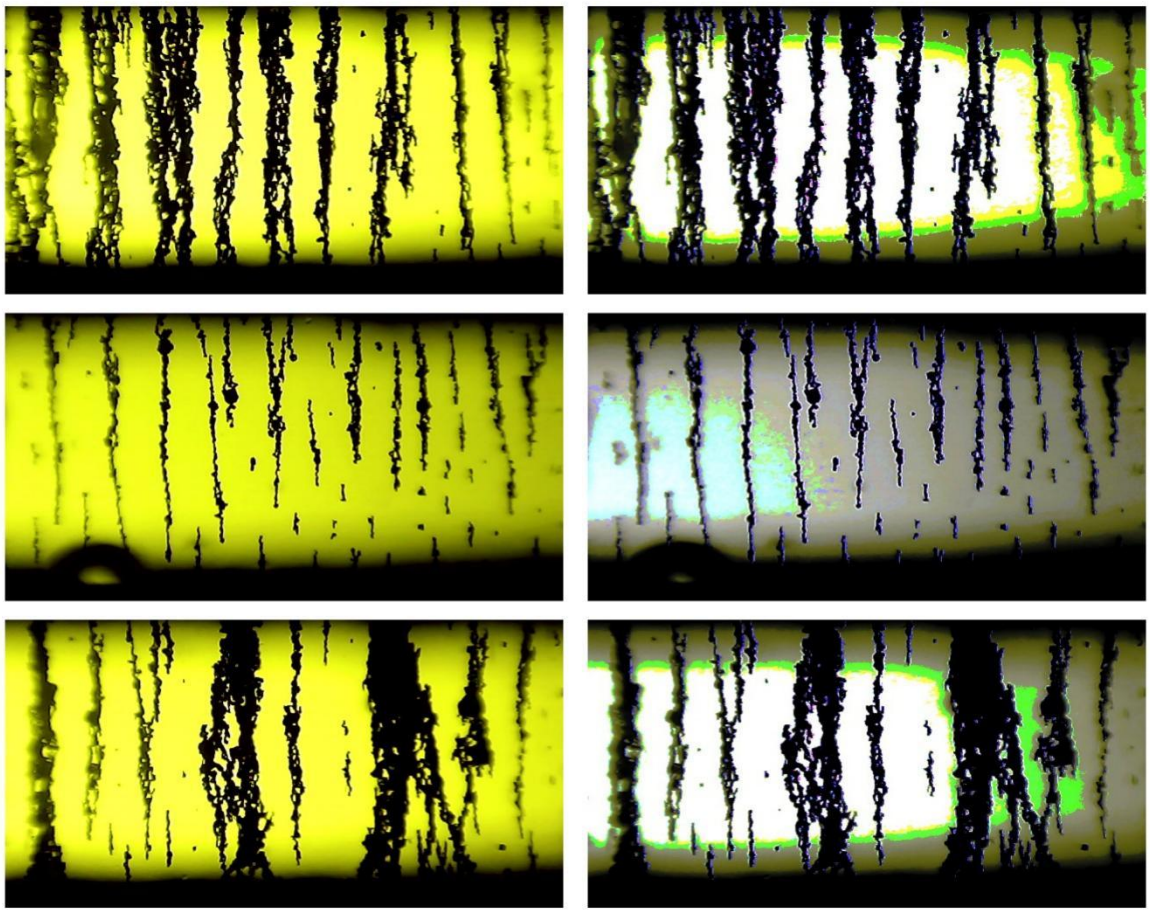
4. lower region of histogram upper region of histogram (Source: [9])

The images are then stacked together to obtain the average of the image pixels between these two images obtained from the lower and upper regions of histogram. With the z- axis as reference, the images are aligned together so that the images are located perpendicular to the z-axis and the image plane itself. The under-enhanced image is then projected with the over-enhanced image. Each pixel of both images is compared at a particular pixel location, and the average value from these pixels is determined. This process produces an improved image in terms of contrast, with the output image having excellent contrast [9]. The output image is then converted into HSV color model [11].

3.1.2 Outputs From Rayleigh Stretching and Averaging of Image Planes

Fig. 5 shows the output from Rayleigh Stretching and Averaging of Image Planes. In the figure we can clearly see that, the image contrast has been changed. Also, the resolution at the edges is improved and the wear particles present in blurry areas are also visible. However, the edge definition in blurred areas is not satisfactory. Hence, we had to look for other methods that can be applied on top of the output to improve image properties in blurred areas as well. For wear particle characterization, it is very important to have well defined edges because edges of wear particles hold a prominent about of information regarding the reason for wear.

Upon reviewing, we came across a very famous and effective method built for enhancing properties of blurry underwater images i.e Image Blurriness and Light Absorption (IBLA) method. It is a depth estimation method for underwater scenes based on image blurriness and light absorption, which can be used in the image formation model (IFM) to restore and enhance underwater images [13].



5. original image and its enhanced output from rayleigh stretching and averaging of image planes

3.1.3 Image Blurriness and Light Absorption (IBLA)

In order to improve the image quality in blurred regions and regions where background blends into foreground wear particles, going through the paper mentioned in [13], we found the method most suitable as the transmission map from IBLA is capable to produce image that can represent 98% of the world's open ocean and coastal waters fall.

IBLA restores an image in four different steps.

- Image Blurriness Estimation
- Background Light Estimation
- Depth Estimation Based on Light Absorption and Blurriness
- Transmission Map Estimation and Scene Radiance Recovery

As the paper focusses on the development of a denoising and background removing refined and revised U-Net Architecture, we hence, will view all these steps in brief.

The blurriness estimation was first presented in the paper [14]. It includes three steps. Let $G^{k,\sigma}$be the input image filtered by a k ×k spatial Gaussian filter with variance σ^2. The initial blurriness map P_{init} is computed using (6) [13]:

$$P_{init}(x) = \frac{1}{n}\sum_{i=1}^{n}|I_g(x) - G^{r_i,r_i}(x)|$$

Where I_g is the grayscale version of the input image I c, $r_i = 2^i n + 1$, and n is set to 4.

Next, we apply the max filter to calculate the rough blurriness map P_r as in (7) [14]:

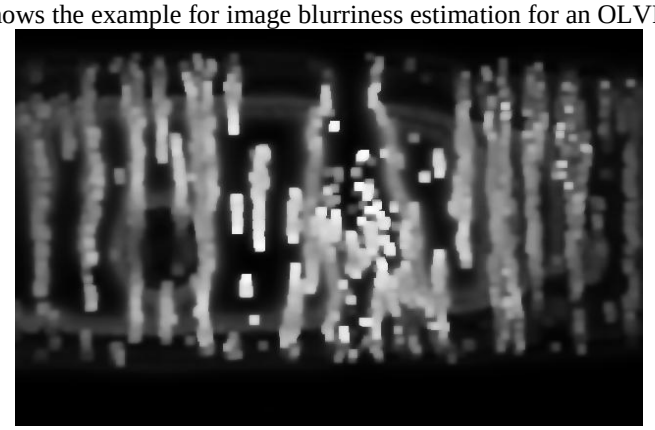
$$P_r(x) = \max_{y \in \Omega(x)} P_{init}(y)$$

Where Ω(x) is a z × z local patch centered at x. Here, we set z = 7. We refine P_r by filling the holes caused by flat regions in the objects using morphological reconstruction [15], and then soft matting [16] or guided filtering [17] is applied for smoothing to generate a refined blurriness map P_{blr} as in (8) [14]:

$$P_{blr}(x) = F_g\left\{C_r\left[P_r(x)\right]\right\}$$

Here C_r is a hole-filling morphological reconstruction operator, and F_g is the soft matting or guided filtering function.

Fig. 6 shows the example for image blurriness estimation for an OLVF image.



6.image blurriness estimation

The next step involves the background light (BL) or atmospheric light estimation. BL determines the color tone of an OLVF image as well as its restored scene radiance. For an OLVF image, the lower and upper bounds of its possible restored scene radiance $J_c \in [0, 1]$ can be derived by setting $B_c = 1$ and $B_c = 0$ in (9) [13]:

$$\max\left(\frac{I^c - 1 + \tilde{t'}}{\tilde{t'}}, 0\right) \leq \tilde{J^c} \leq \min\left(\frac{I^c}{\tilde{t'}}, 1\right)$$

where,

where $\tilde{t} = \max(\tilde{t}^c, t_0) \in [t_0, 1]$

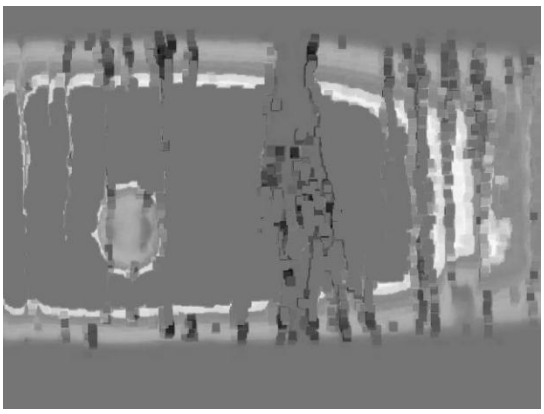
$I^c(x) = J^c(x)t^c(x) + B^c(1 - t^c(x))$, $c \in \{r, g, b\}$

$$\tilde{J^c}(x) = \frac{I^c(x) - \widetilde{B^c}}{\max\left(\tilde{t^c}(x), t_0\right)} + \widetilde{B^c}$$

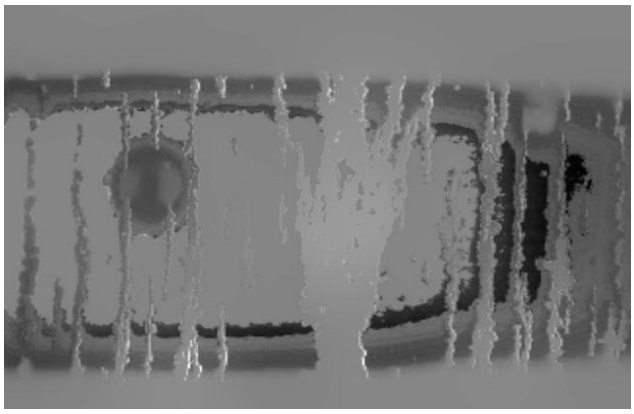
In between these extremes, the BL estimate is calculated by a weighted combination of the darkest and brightest BL candidates. The python code for the entire process is taken from [18].

After the background light estimation between the two extremes, we then estimate the depth of the image. The depth estimation is carried out over each individual channels as described in [14]. Fig. 7 shows the example of depth estimation for an OLVF image.

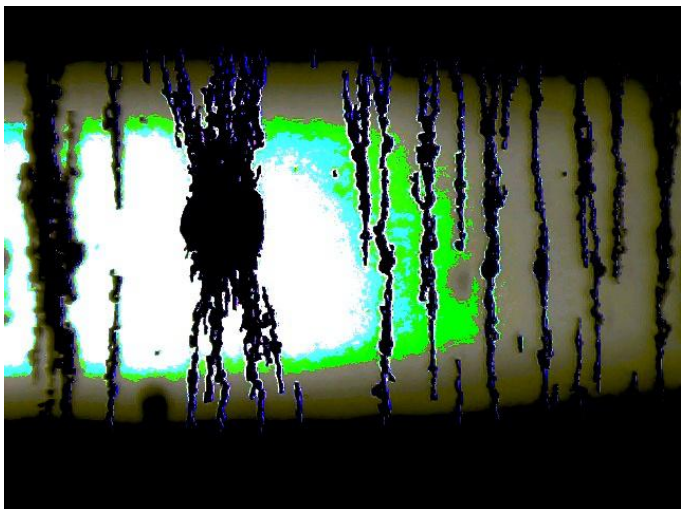
As a final step, we compute the Transmission Map and the Final Scene Radiance as described in [13] and used the code available in [18] for the process. Fig. 8 and 9 show the example of Transmission Map and the Final Scene Radiance.



7. depth estimation



8. transmission map estimation



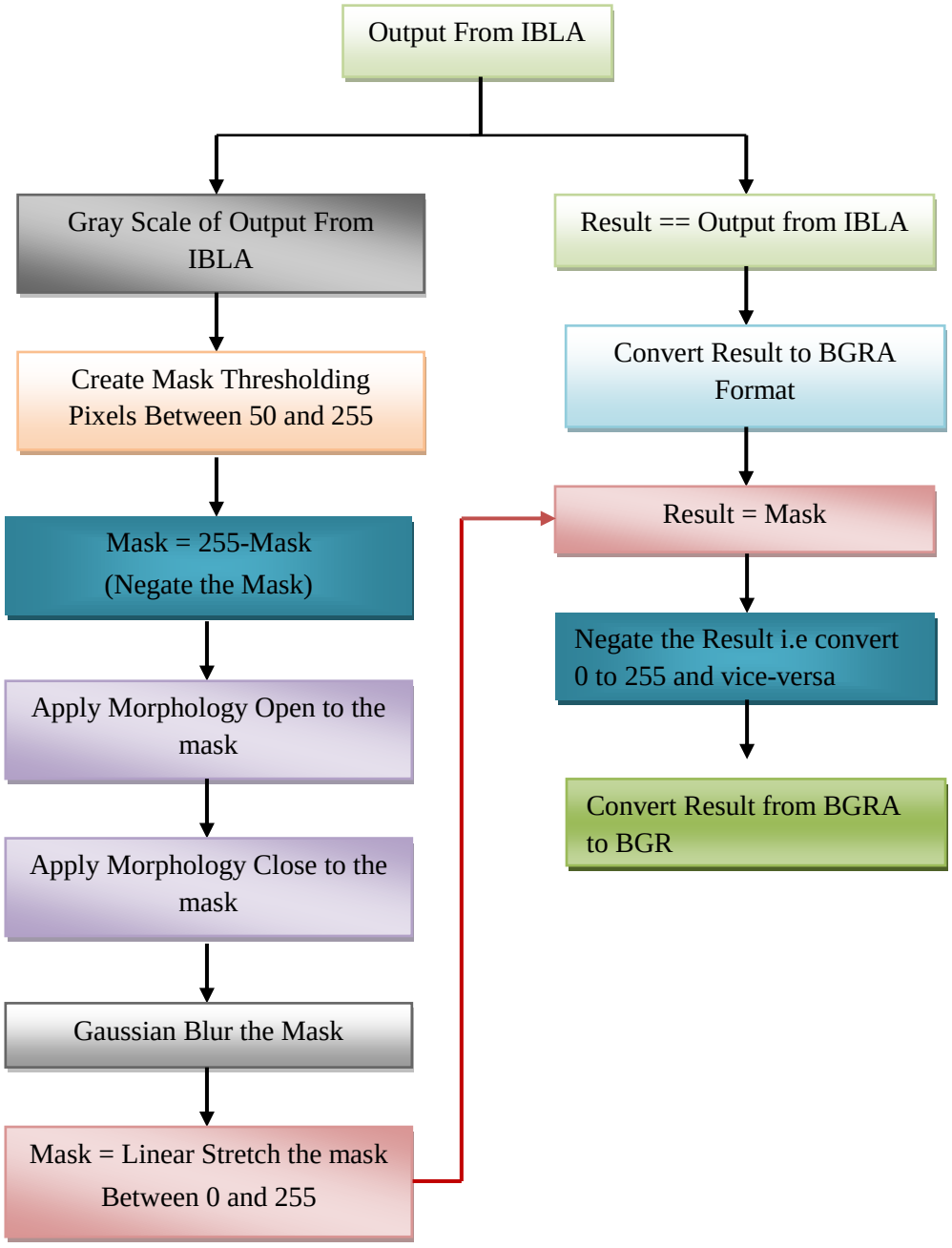
9.output after ibla on rayleigh stretched image

4. Background Removal

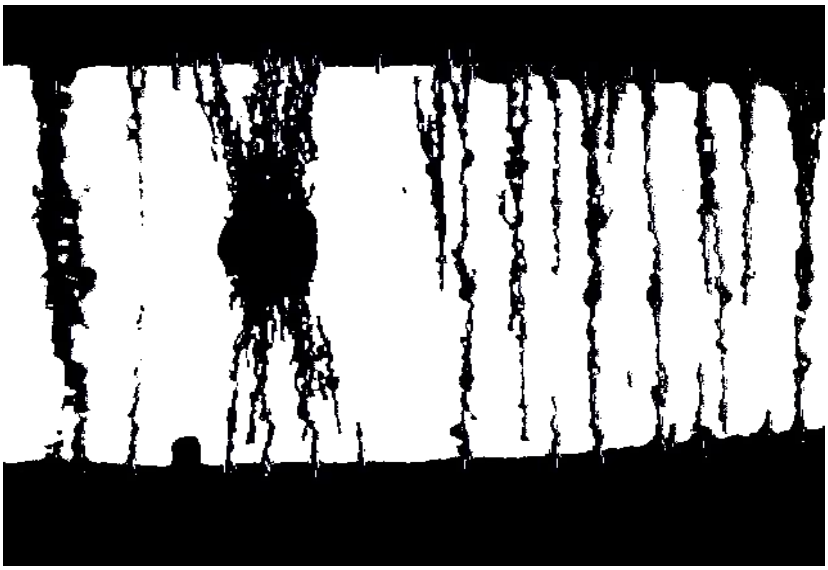
The enhanced image obtained after Rayleigh Stretching and Averaging of Image Planes and IBLA is then taken for background removal. After the image processing, the foreground image has a different color compared to its background. Also, most of the edges are defined well after the procedure thus, background removal became easier.

As an initial step to the process, the image is converted into grayscale and to convert the grayscale image into binary image, thresholding is done setting a pixel range of 50 to 255 to form a mask. Then, we negate the entire mask by subtracting each pixel by 255. After that, in order to remove any form of noise in the binary image, we then apply erosion followed by dilation. This process is also called as morphology open. Then to close the small holes present in the foreground image of the mask, we then carry out the opposite process i.e dilation followed by erosion. This process is called morphology close.

After closing the small holes, we then do the Gaussian Blur on the transformed image and then we carry out the linear stretching of the image where the image is stretched within the range of 0 to 255. Now, we apply the linear stretched mask over the original image by converting the original image into a BGRA format. Now, as our linear stretched output is negative, to convert it into positive image, we then assign 255 to pixels with pixel value of 0 and finally convert the output to a BGR format. Fig. 10 and 11 show the steps involved during background removal and the output obtained after background removal respectively.



10. background removal process



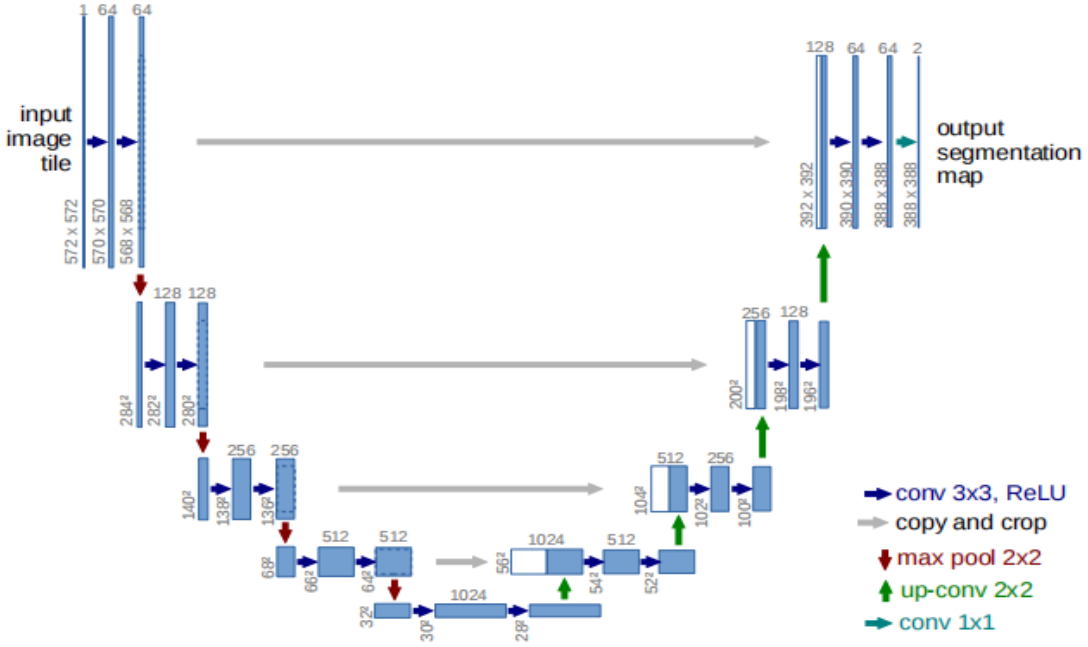
11. final output after background removal

5. Proposed Method: Zoomed-Fired U-Net (ZF U-Net)

There is large consent that successful training of deep networks requires many thousand annotated training samples. However, most of the analysis and research going on within the scientific community have very less number of available annotated training dataset. In such scenario, the U-Net architecture gave a silver lining and paved a way of strong data augmentation and helped counter the problem.

The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization [19]. The network architecture uses a 3x3 kernel for convolution operations and a 2x2 kernel for max pooling. The convolution operation helps the network to learn filter/ kernel values that can produce the most optimal weights for that particular network parameters and the max pooling down samples the input along its spatial dimensions (height and width) by taking the maximum value or an average over an input window (of size defined by pool size) for each channel of the input [20].

Fig. 12 shows the U-Net architecture as proposed in the original paper.



12.u-net architecture (source: [19]**)**

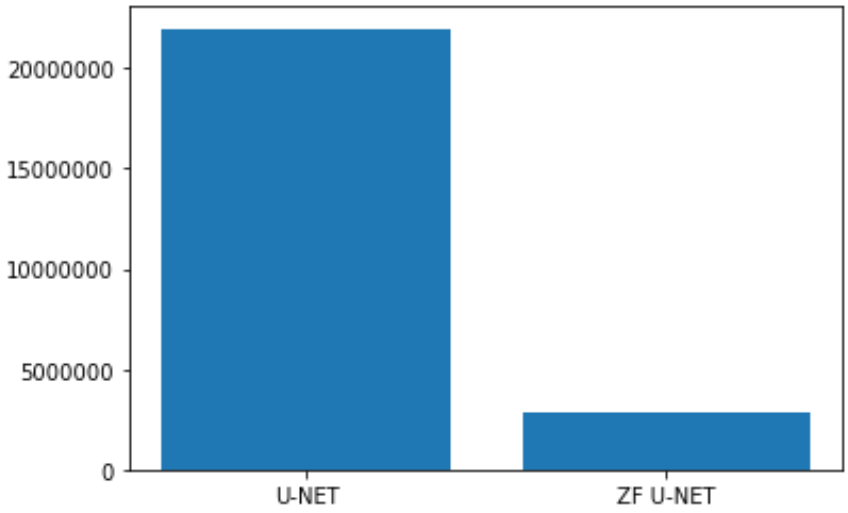
Here, in the figure, each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

U-Net architecture performed well for the specified operations but as the world is focusing more on usage of deep learning approaches for real-time operations, it is a must to decrease the model parameters and also make the network as light as possible to improve the speed of the model without compromising the accuracy.

In order to achieve that, we proposed our ZF U-Net that can give speedy results without loss in accuracy or a negligible loss in accuracy for our scenario of OLVF image denoising and background removal. For the training our network, we used the dataset obtained as explained in dataset preparation as our targeted output.

5.1.1 Performance Comparison With U-Net

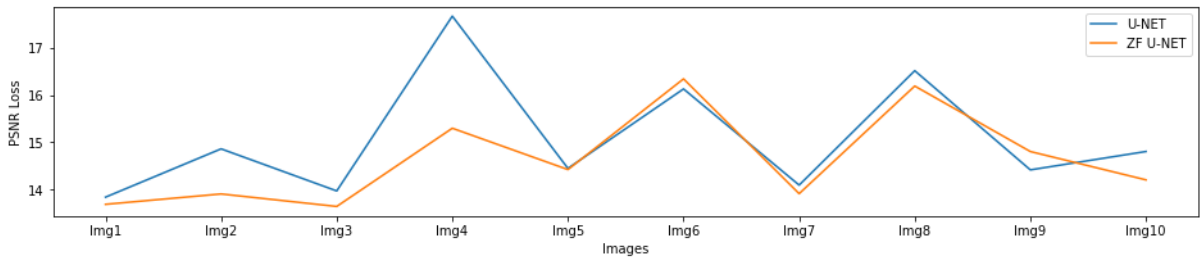
We compared performance of our network with U-Net based on number of model parameters to be updated during training, the time taken for prediction and the PSNR loss. Fig. 13 shows the total number of trainable model parameters in U-Net and our proposed model ZF U-Net.



13. comparison of number of trainable parameters in u-net and zf u-net

The bar plot clearly shows that ZF U-Net has more than 80% less parameters compared to U-Net Architecture. A less dense network has often improved speed but there is a risk of loss of accuracy since model may fail to learn some of the hidden features present in the images.

In order to compute the model performance based on accuracy, we preferred PSNR loss and computed the PSNR loss between the output from both the models and compared it with their respective targeted outputs. The maximum PSNR value expected between the output from the model and the targeted output is 30. In order to convert the problem into a minimization problem, we subtracted the PSNR loss at the model output by 30 so that the model focusses on minimizing that particular value and plotted Fig. 14 showing the PSNR losses for the two models.

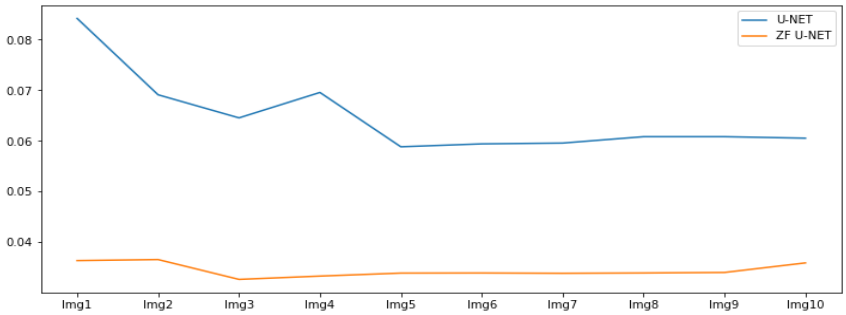


14. comparison of psnr losses in the output by u-net and zf u-net

As the loss now is a minimization problem, the less the value, the better the model performance. So, we can see that for Img2, Img4, ZF U-Net performs much better and in other scenarios as well, the model performs

almost equivalent to U-Net. So, despite having more than 80% less parameters, the model's loss is almost equivalent to U-Net and in some scenarios even excelled the U-Net performance.

The third parameter for comparison of model performance is the time to prediction. Fig. 15 shows the time taken for prediction by both the models.



15.comparison of time taken for prediction by u-net and zf u-net

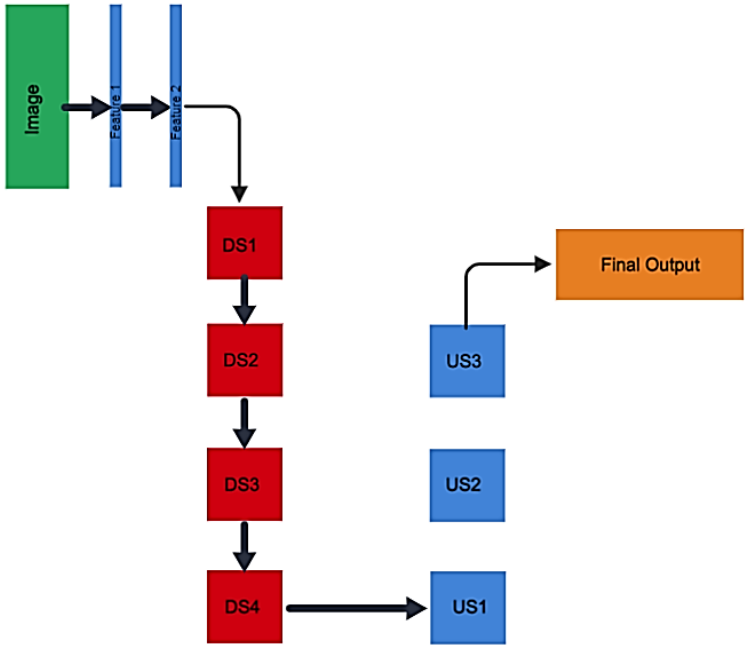
The graph clearly shows that U-Net takes quite a while to predict some of the images and stabilizes around 60 ms and our proposed model is almost twice fast as U-Net and is capable to predict within almost 30-35 ms. Thus, the above graphs clearly represent that the proposed model excelled U-Net in multiple aspects for OLVF image denoising and background removal.

6. ZF U-Net Architecture

ZF U-Net is a revised and updated model similar to U-Net. Just like U-Net, ZF U-Net also has a contracting and expanding pathway which include down sampling and up sampling. The down sampling as the name suggests is the processing of decreasing the size of feature vectors to increase computational speed however this step results the loss of information. Down sampling helps to thin the network and thus makes the process memory efficient and hence improve the model's speed.

The opposite of down sampling is up sampling. Here in up sampling, we increase or add the information to the feature vector. U-Net gains the information for up sampling from their down sampling layers at the same level. However, ZF U-Net gets the information for it's Up sampling in a criss-cross manner, so that any possible overfitting and linearity could be broken down.

Fig. 16 shows the basic architecture of ZF U-Net without skip connections.



16. basic zf u-net architecture

Here in figure, the feature 1 and feature 2 represent the features extracted from the image. DS represent down sampling and US represent up sampling.

6.1 Down Sampling

The down sampling has a zoom module followed by fire module. In the zoom module, we carry out the Bilinear Interpolation based zoomed convolution operation and in fire module, we carry out 1x1 and 3x3 convolution followed by concatenation.

6.1.1 Bilinear Zoomed Convolution

In Bilinear Zoomed Operation, we resize the image feature map to 80% of it's previous feature layer using the bilinear interpolation operation. In order to achieve this, we have to find the average value between two pixels. After that we use that average value as the pixel value between those two pixels. This can be applied and completed for the rows. Thus, we obtain a new expanded matrix. After the expansion is done row wise next step is done by column expansion in the same way[21].

After the expansion, we carry out convolution operation on the bilinear interpolated matrix followed by LeakyReLU based activation and again followed by bilinear interpolation to resize the matrix to the same shape and size as of the previous layer.

So, the entire process can be summarized as (10): For a previous layer $P_{(x,y)}$:

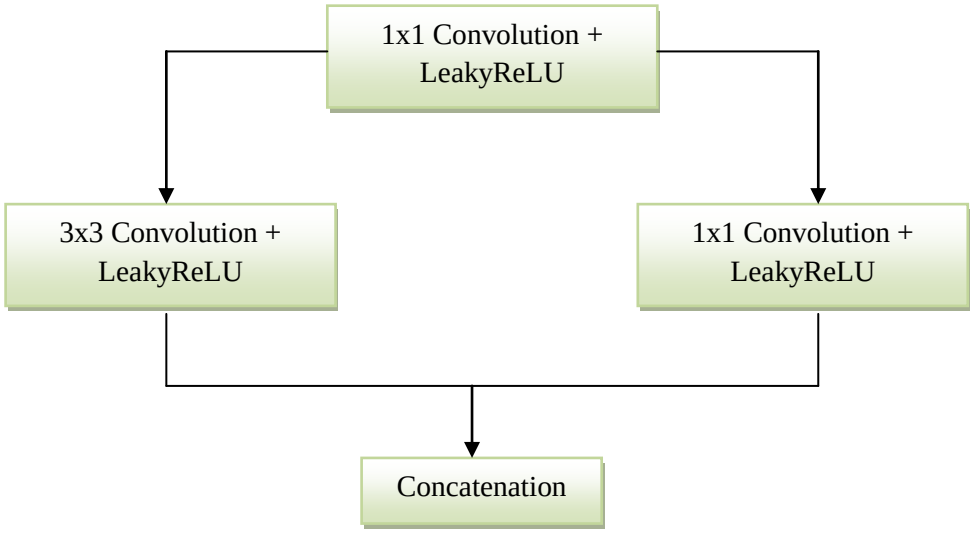
$$Z_{(x1,y1)} = B_{i(x,y)}[C[B_{i(x,y)}[P_{(x,y)}]]]$$

Here, $B_{i(x,y)}$ represents the Bilinear Interpolation Operation and C represents the convolution and activation operation.

6.1.2 Fire Module

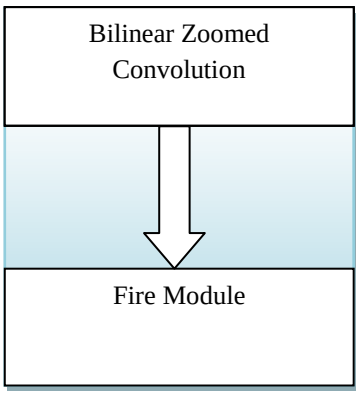
The concept of fire module is extracted from the paper [22]. A Fire Module has a 1x1 convolution operation done on the previous layer which is followed by 1x1 convolution and 3x3 convolution operation on the output from the 1x1 convolution operation done initially. The outputs from the 1x1 convolution and 3x3 convolution are concatenated together to produce a final output. Unlike U-Net architecture, which mostly relied on 3x3 kernels for convolution, the 1x1 convolution operation made the computation much lighter and helped in speeding up the process.

A 1x1 convolution operation helps to down sample the depth of feature maps. A 1×1 filter will only have a single parameter or weight for each channel in the input, and like the application of any filter results in a single output value. This structure allows the 1×1 filter to act like a single neuron with an input from the same position across each of the feature maps in the input. This single neuron can then be applied systematically with a stride of one, left-to-right and top-to-bottom without any need for padding, resulting in a feature map with the same width and height as the input [23]. Fig. 17 shows the process involved in Fire Module.



17.fire module operation

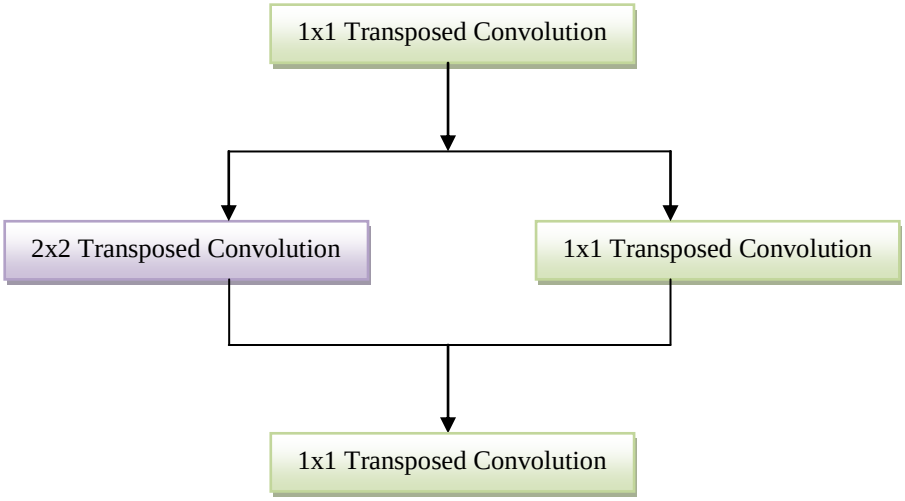
The entire down sampling operation can be presented as shows in Fig. 18.



18.down sampling process

6.2 Up Sampling

For up sampling, we have used the concept of transposed fire module from the paper [22]. Our transposed fire module has a 1x1 transposed convolution followed by LeakyReLU whose output is fed into a 2x2 transposed convolution network and a 1x1 transposed convolution network. The outputs from both the networks are then concatenated to produce a final output. Fig. 19 shows the up sampling process.



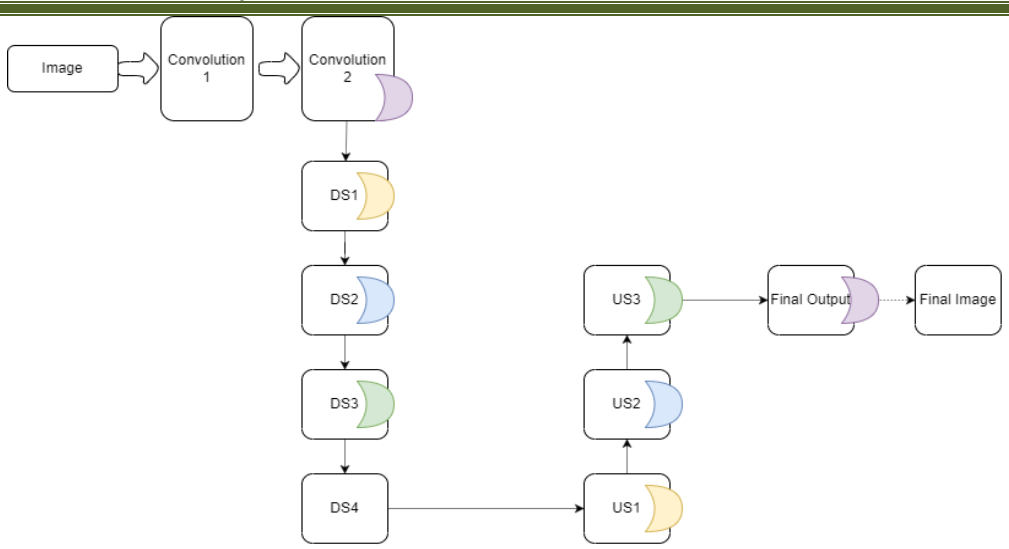
19. up sampling process

6.3 Skip Connections

Skip Connections (or Shortcut Connections) as the name suggests skips some of the layers in the neural network and feeds the output of one layer as the input to the next layers. Skip Connections were introduced to solve different problems in different architectures. In the case of ResNets, skip connections solved the degradation problem, in the case of DenseNets, it ensured feature reusability [24]. Here, just like in U-Net Architecture, we have made use of DenseNets for feature reusability.

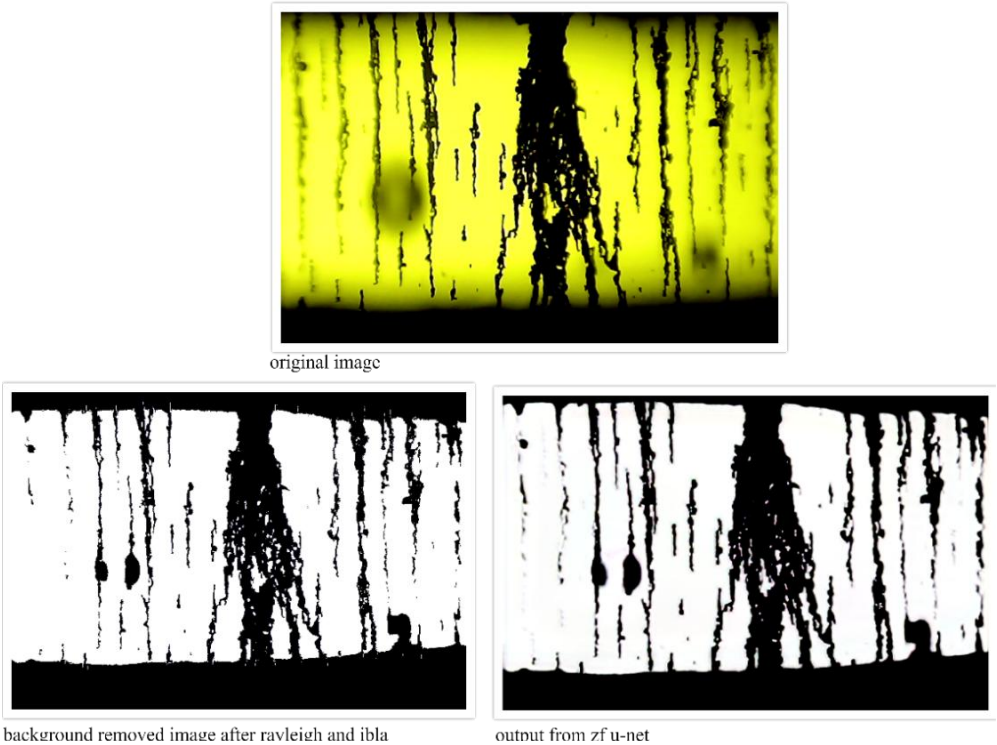
U-Net uses skip connections to concatenate features in contracting and expanding paths lying at the same level. In ZF U-Net, we have made use of zig-zag pattern so that the feature maps extracted at different levels could be cross-linked. This helps to break any existing linear patterns between the feature maps.

So, finally our network architecture looks as shows in Fig. 20.



20. zf u-net architecture

The above figure shows the complete ZF U-Net architecture. The image is passed through a set of convolutional neural networks followed by subsequent down sampling and up sampling layers. Every convolution operation is followed by an activation function LeakyReLU. The crescent symbols with similar colors are to be skip connected. For example, As DS1 has same crescent color as US1, hence, we are going to concatenate the feature maps of US1 and DS1 together to form an output that is fed to US2. For the final layer i.e. the final output, the output from Final Output and Convolution 2 are concatenated to form the final image. Fig, 21 shows the final output from the network.



21. outputs from different processes

6.4 Loss/ Cost Function

Any supervised network architecture trains on the basis of difference between the expected output and the output from the network. The difference can be calculated using different cost functions/ loss functions. Here, for ZF U-Net for OLVF image denoising and background removal, we have used the PSNR Loss as the loss function. The PSNR loss function is converted into a minimization problem by setting the maximum

expected PSNR value as 30. The model then tried to reduce the difference between the expected maximum PSNR value and the value obtained at the output.

7. Unsupervised Clustering of Wear Particles

Any wear model of a mechanical system can be developed only after understanding the types of wear particles coming from the system. So, individual categorization or classification of wear particles through manual methods is highly tedious and the number of particles that could be classified would be limited. So, if we could get some idea on what type of wear particles fall into what category then, it would be easier to develop a true wear representation of the system. For that, the only option left is the use of unsupervised learning method which has performed better in prevailing annotated dataset irrespective of the domain.

The most prominent problem in the field of wear particles classification using deep learning is the lack of availability of annotated data to train a classification model. This challenging problem is a major hindrance for our research work which also focusses on the development of a deep learning model capable of classifying wear particles extracted from the denoised and background removed OLVF image obtained after processing with ZF U-Net.

In the lack of the availability of the annotated data, the only solution to the problem is either manual annotation of every wear particle or use of unsupervised clustering techniques. For our problem, we opted the latter i.e. the use of unsupervised clustering techniques.

For clustering of wear particles into individual clusters based on their features, the first basic and important step to proceed further is the extraction of wear particles from individual image obtained from ZF U-Net. The manual extraction of wear particles present in the image is highly tedious and consumes much time and energy. Not only that, the manual extraction is based on human judgement which sometimes fails to analyze the hidden features that many be present in the image. Thus, here in this paper we made use of selective search algorithm to extract wear particles or propose 500 regions of interest (ROIs) from a single image.

After the extraction is done, we then have to check whether the extracted wear particles are duplicates or whether the intersection over union (IOU) of the wear particles crosses certain threshold limit since we don't want our network to learn the similar features again and again. If the IOU of two extracted image exceeds the threshold value, we then count the number of the white pixels in the images and select the one with less white pixels. We then perform unsupervised clustering using Semantic Clustering by Adopting Nearest Neighbors to group images with similar features into one cluster. Fig. 21 shows the entire process involved during the clustering process.

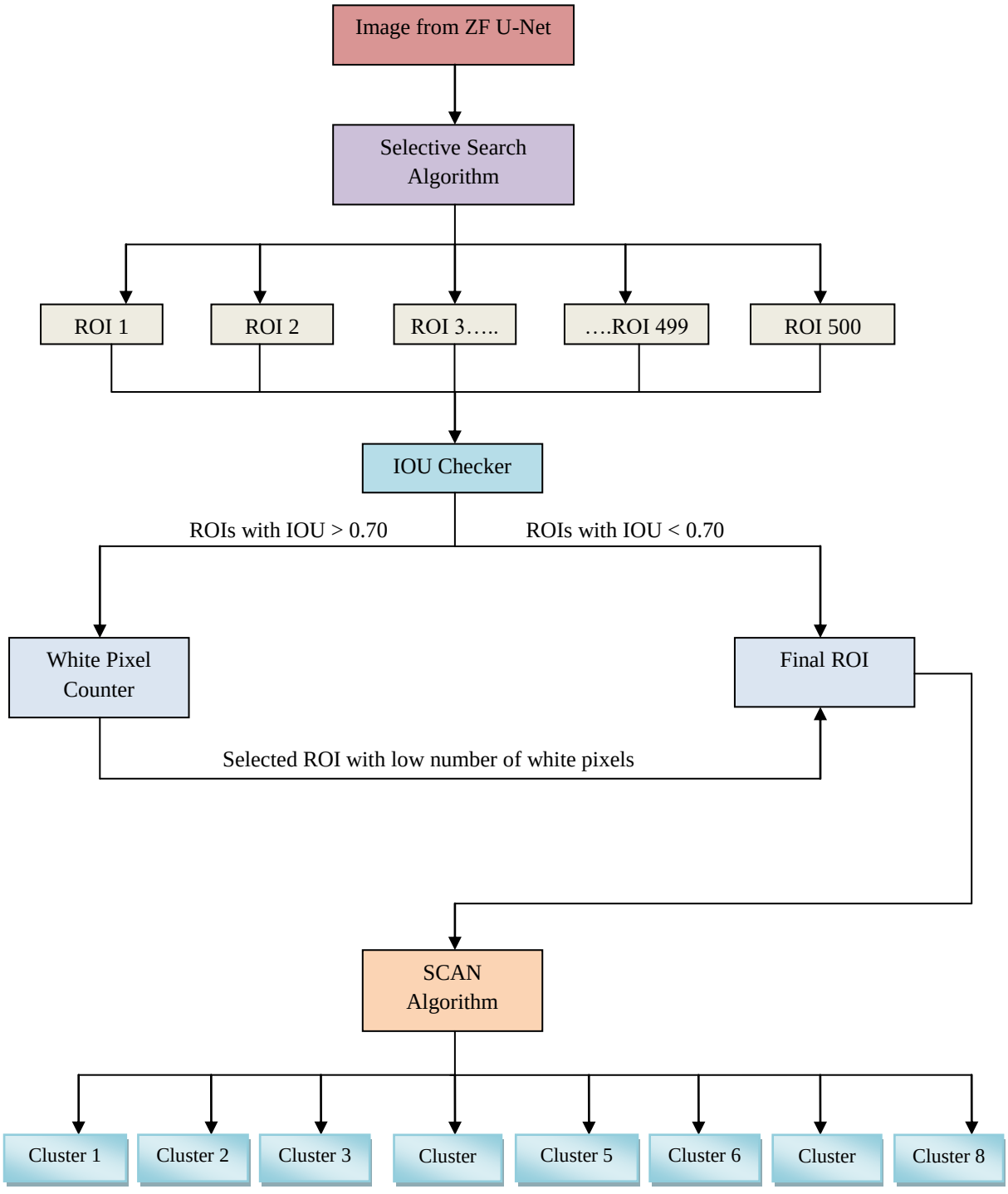
7.1 Proposing ROIs Using Selective Search Algorithm

Manual extraction of individual wear particles from the denoised and background removed image is a tedious task and also is an impossible task when the number of wear particles in an image goes really high. So, automating the process helps to save a lot of time. For the process, we have made use of selective search algorithm which is based on multiple grouping criteria like color, texture, shape, brightness etc. It is capable of detecting objects at any scale. It uses bottom-up grouping of image regions to generate a hierarchy of small to large regions [25].

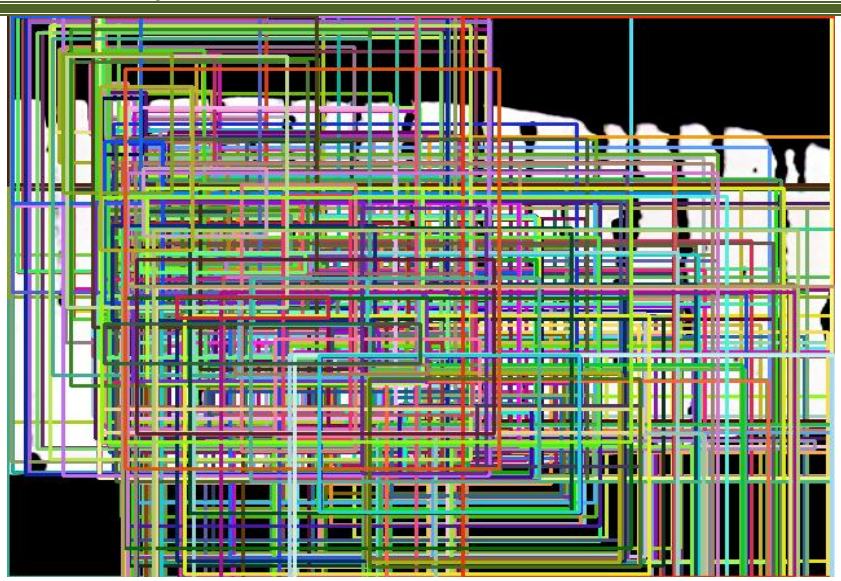
Selective Search Algorithm is the combination of both the exhaustive search method and segmentation. It captures all the scales, addressing the fact that objects in images can be present at different sizes and orientations. This step is based on the intuition of the hierarchical structure of images. The creation of initial regions is done via a graph-based greedy algorithm, which starts grouping the most similar regions until the whole image becomes a unique region.

It diversifies the grouping of regions according to different metrics. Indeed, there is no optimal strategy to group regions together: sometimes might be because of the color, sometimes because of the texture, and so forth. Specifically, the authors introduced four different diversification strategies based on color, texture, size and fill. For each feature, a similarity score (between each couple of regions) is computed. The final similarity score is a linear combination of the above four similarity scores [26]. For our cases, we just considered 500 regions of interests proposed by the algorithm to proceed further. Fig. 22 shows the 500 regions of interest proposed by the selective search algorithm.

Each individual rectangles proposed is our region of interest (ROI). Now, ever ROI proposed is then passed through an algorithm called Intersection Over Union (IOU) which computes whether the individual ROI proposed represents a unique wear particle. IOU of an ROI is computed with all other 499 remaining ROIs to ensure that the information present in it is unique.



21. proposing region of interests and clustering process

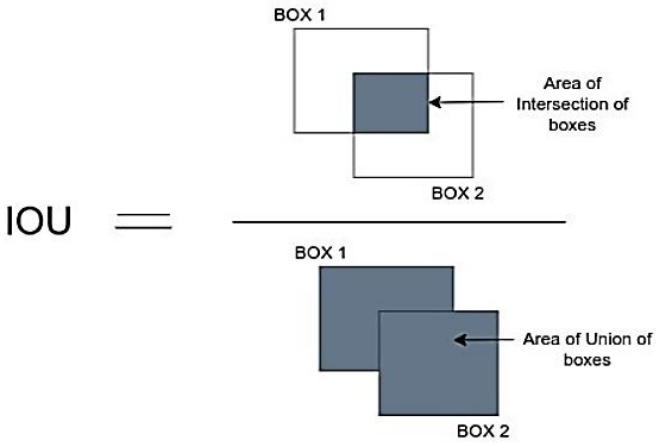


22.output from selective search algorithm

7.2 Computation of Intersection Over Union (IOU) and White Pixels Counting

IOU is a term used to describe the extent of overlap of two boxes. The greater the region of overlap, the greater the IOU. IOU is mainly used in applications related to object detection, where we train a model to output a box that fits perfectly around an object. However, for our case we are using IOU to confirm that each of the bounding boxes obtained from the selective search algorithm represent a unique wear particle.

Fig. 23 shows the way to calculate IOU of two boxes Box 1 and Box 2. We have set a threshold of 70% i.e. if two boxes have an IOU greater than 70%, then we consider the two to be representing the same wear particle. So, in order to select one box from the two, we then use the concept of white pixel counting. The box with least number of white pixels is selected since the information about the wear particle is more in that particular box and thus is considered as the final region of interest (ROI).



23. calculation of iou

7.3 Unsupervised Clustering of Proposed ROIs Using SCAN Algorithm

As a final step for the research, we pass the proposed ROIs to an unsupervised algorithm called SCAN which clusters them into individual clusters based on their similarity and differences in the features. The reason behind selecting the algorithm for the clustering are:

- The proposed method eliminates prior knowledge requirements of:

(a) ground-truth semantic labels at training time and
(b) the number of classes.

- Strong Augmentation of data helps in improving model performance.
- Model is not much dependent on other factors like number of neighbors, cluster estimation and even the network architecture selected for representation learning.

For training the SCAN algorithm in our case, we used 129,404 ROIs as training images for the algorithm and for testing we used 69,680 ROI images. We considered the following parameters during the training process:

1) Number of Neighbors = 10
2) Representation Dimension = 768
3) Projection Units = 256
4) Number of Clusters = 8

As the model performance is highly dependent on data augmentation, we considered four different augmentation methods for our images. They are:

1) Random Translation
2) Random Flip
3) Random Rotation
4) Random Zoom

7.3.1 The Model Architecture

The SCAN algorithm is a combination of self-supervised learning followed by unsupervised clustering processes. It involves the following processes to cluster images into their respective clusters.

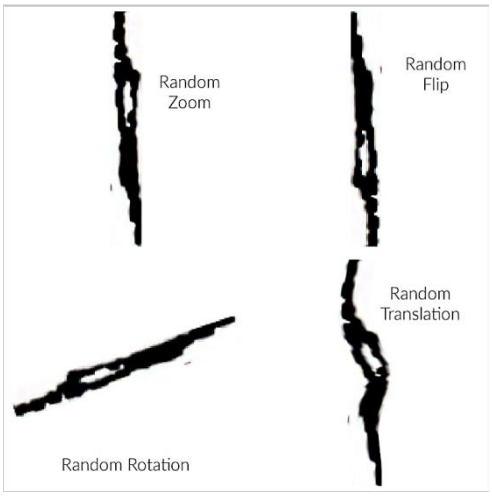
7.3.1.1 Representation Learning

Feature learning or Representation Learning is a technique that enables a system to learn the features from a raw data automatically so that it can be utilized for classification or feature detection. It extracts hidden information present in the image or any raw data that may or may not be visible to human eyes. In case of supervised learning, in order to generate the best feature learning network, generally cross-entropy losses (for classification problems) are used as a metric to evaluate the difference in the expected output and the output from the network. The output and the expected outputs can be classes or a feature vector. On the basis of the difference, also called as model loss, the model updates it weights so that the loss can be minimized.

During the initial phase of training a network, a network always learns low-level features like contour, texture, shape, color etc. and the model can overfit into low-level features and can consider them as the only basis for classification or clustering. To overcome these limitations, SCAN algorithm employs representation learning as a means to obtain a better prior for semantic clustering.

In representation learning, a pretext task (ζ) learns an embedding function (φ_θ) parameterized by a neural network with weights θ- that maps to feature representations. Since, we don't have any labels assigned to the proposed ROIs, we don't have any means to confirm whether the feature or embeddings from the network is accurate. In order to counter the problem, SCAN algorithm employs the concept of data augmentation or image augmentation.

Being based on the idea proposed in the SCAN original paper [27], we employ the concept of data augmentation whether we produce four different images from a single ROI using Random Translation, Random Flip, Random Rotation and Random Zoom. Fig. 24 shows the image augmentation techniques used in the research.



24.image augmentation

Image augmentation helps to produce the best pretext task function. A pretext task function or an embedding function learns by minimizing the distance between features embedding of the original image (X_i) and its augmented image $T[X_i]$ (For our case, Random Zoom, Random Flip, Random Rotation and Random Translation). Hence, the function can be written as in (11):

$$\min_{\theta} d(\Phi_\theta(X_i), \Phi_\theta(T[X_i])).$$

Fig. 25 shows the total parameters involved for representation learning for our case.

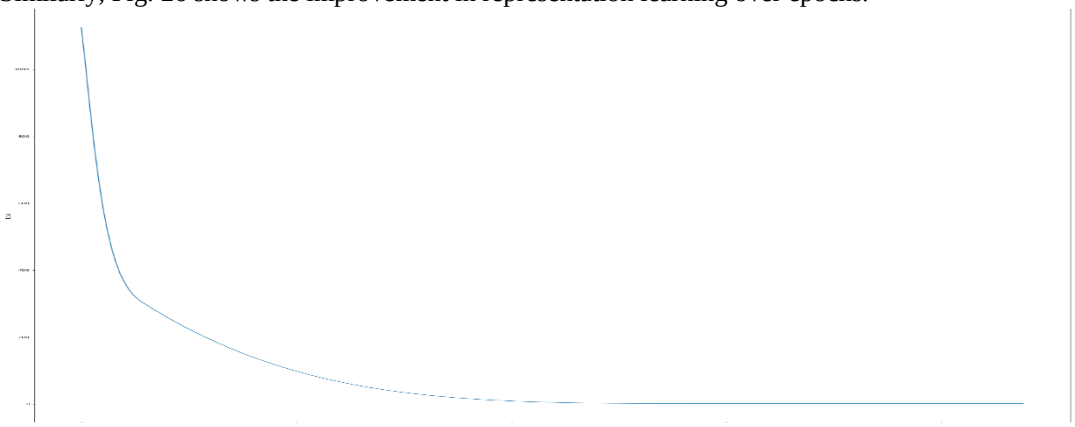
```
Model: "representation_learner"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential_1 (Sequential)   (None, 768)               25138432

 sequential_2 (Sequential)   (None, 256)               197632

=================================================================
Total params: 25,336,066
Trainable params: 25,290,112
Non-trainable params: 45,954
```

25.number of parameters involved for representation learning

Similarly, Fig. 26 shows the improvement in representation learning over epochs.

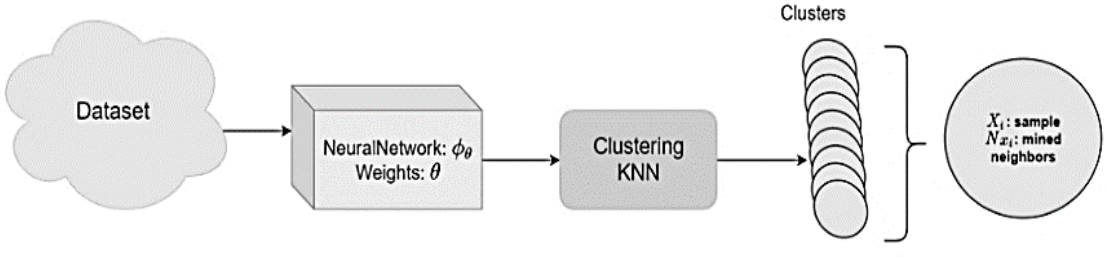


26. reduction in loss over epochs

The representation model was trained for an epochs of 250 to reduce the loss to a value of 1.38.

7.3.1.2 Mining Nearest Neighbors

A representation learning model, φ_θ computed the most suitable vectors for the proposed region of interest. Now, the vectors need to be clustered into same or different clusters based on their feature similarity and differences. For every image in the training dataset, neighbors having close features to it is computed using K-Nearest Neighbors Algorithm. Here for our case, we are considering 8 different clusters into which the images are to be segregated and we are going to consider 10 different feature vectors before considering that particular feature vector to lie in that particular cluster. In order to compute the closeness of the feature vector from other neighboring feature vectors present in each cluster, we are adopting the Euclidean Distance Method. Fig.27 shows the entire mining process.



27. mining nearest neighbors

7.3.1.3 Clustering: A semantic clustering loss

From the step explained above we have X_i and its mind neighbors N_{xi}. Now, in this step, we are training a neural network which classifies X_i and N_{xi} into same cluster. The clustering function or the network is $\Phi\eta$ parameterized by weights η. As a final step for the function, a softmax activation function is used to predict the cluster in which X_i is to be assigned. For our case, number of cluster is set to 8. So, the softmax function gives a confidence value for every image assigned to every cluster.

The loss function used to update the weights is given by (12) [27].

$$\Lambda = -\frac{1}{|\mathcal{D}|}\sum_{X\in\mathcal{D}}\sum_{k\in\mathcal{N}_X}\log\langle\Phi_\eta(X),\Phi_\eta(k)\rangle + \lambda\sum_{c\in\mathcal{C}}\Phi_\eta'^c\log\Phi_\eta'^c,$$

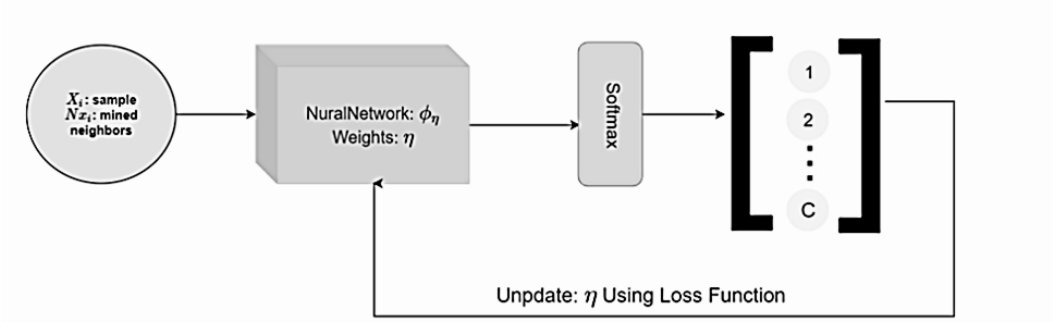
$$\text{with } \Phi_\eta'^c = \frac{1}{|\mathcal{D}|}\sum_{X\in\mathcal{D}}\Phi_\eta^c(X).$$

where, in the loss function, first term, <·> denotes the dot product operator.

The first half of the above equation, tries to minimize the intra-cluster distance i.e. tries to make consistent prediction of X_i and it neighbors N_{xi} into same cluster. This is possible due to the dot product introduced into the equation since the dot product will be maximum only when the predictions are one-hot (confident) and are assigned to the same cluster (consistent).

But trying to be consistent always is a rigidity for the model. This can cluster all the samples into a single cluster. So, to avoid it, a next term entropy is introduced in the equation. The entropy term spreads the predictions uniformly over the clusters. During training of the function or the network, we augment the X_i and the related N_{xi}, so that the network can cluster the X_i and its neighbor into same cluster devoid of any form of orientation or changes which are highly possible in real world scenario.

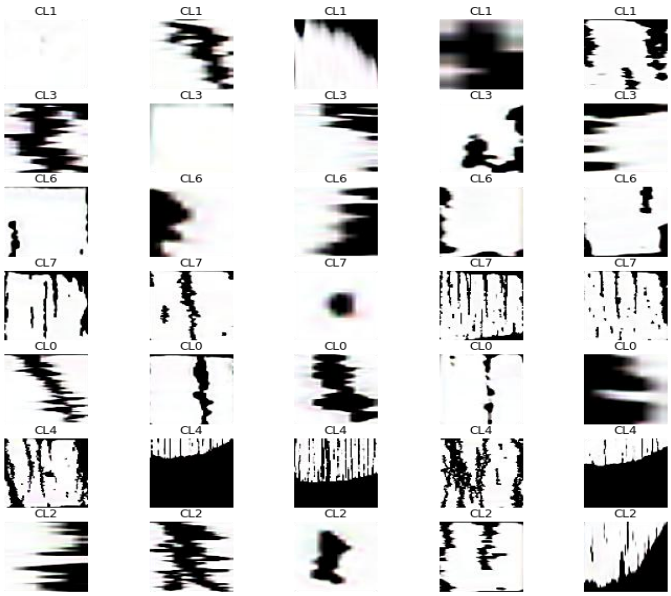
The entire process looks as in Fig. 28.



28. clustering using semantic loss function

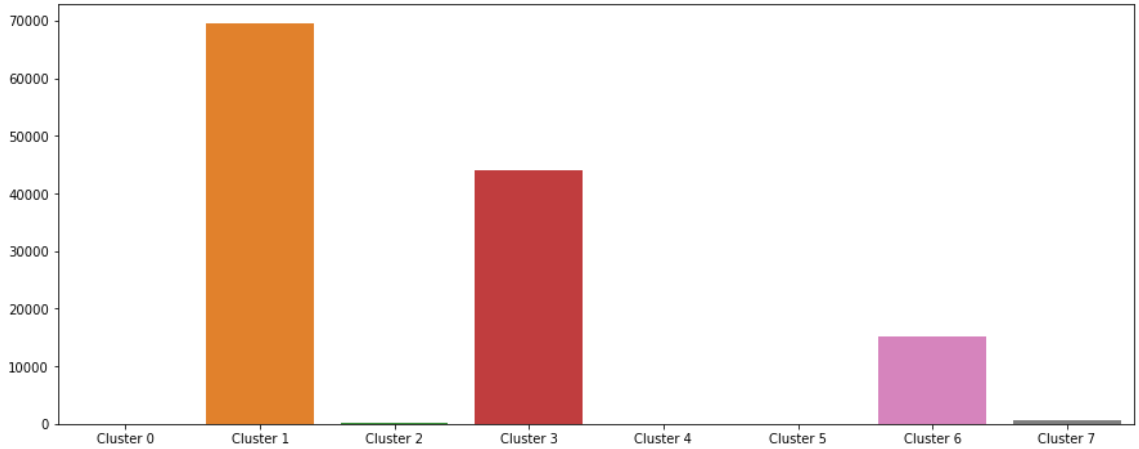
7.3.1.4 Fine Tuning: Self Labelling

Through semantic clustering loss, we were able to bring consistency in clustering process. But it is inevitable that some of the dissimilar samples may be clustered into same cluster. This is mostly the case with samples clustered with a low confidence value. However, there are samples which are clustered into that particular cluster because of having a high confidence of belonging to the cluster. These samples can be called as prototypes for that particular cluster. So, to improve the model performance, we assign pseudo labels to the samples clustered with a high confidence or samples whose pmax> threshold i.e. the pseudo label is the cluster number into which the sample was put into. A cross-entropy loss is used to update the weights for the obtained pseudo labels. To avoid overfitting, we calculate the cross-entropy loss on strongly augmented versions of the confident samples. The self-labeling step allows the network to correct itself, as it gradually becomes more certain, adding more samples to the mix. Fig. 29 shows the output obtained after the entire clustering process where C0 till C7 represent the clusters.

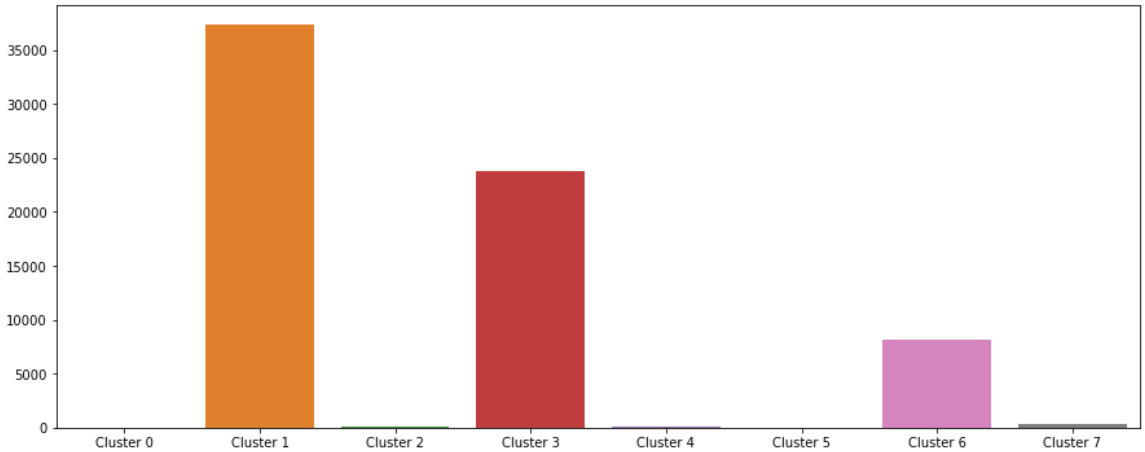


29.clustering of wear particles into individual clusters in test dataset

Fig. 30 and 31 show the number of images classified into each cluster during the model training and prediction over the test images respectively.



30. number of rois classified into each cluster on train dataset



31. number of rois classified into each cluster on test dataset

7.3.1.5 Model Performance

The selected SCAN architecture performed well in different datasets. In the CIFAR-10 dataset, the accuracy with C clusters as mentioned in the original paper was 85.2%. Even after increasing the cluster size twice, the accuracy was 83.8% [28] which confirms that even if we are not aware about the types of wear particles, we can provide a random estimated value for the types of wear particles or the number of clusters and the model has a greater chance to classify them into the right cluster irrespective to the number of clusters assigned.

As the original paper was trained on annotated dataset and the model performance was estimated accordingly, for our scenario, we don't have any annotated dataset to check the performance. However, to get some insights about the model performance, during the step of the selection of region of interest, we knowingly didn't consider the aspect ratio of the proposed region of interest. In Fig. 29 we can clearly see that cluster 4 and cluster 7 represent such images which have unusual aspect ratio in comparison to the other wear particles and the network has segregated them well from rest of the other images.

Along with that, despite no images clustered into cluster 4 and just small number of images clustered into cluster 7 during training of the model, for the test dataset, we can see that the network has clustered some images into cluster 4 and some into cluster 7. However, the images in cluster 7 seem to have some information regarding the wear particles which the model seems to have missed. However, this shows the difference in information present in the two cluster.

This model has given some hands for upcoming researches so that we don't have to jump blindly for data annotation of wear particles in future days. Along with a human evaluator and the model, it would be easier to annotate the wear particles so that we can develop a more reliable model which can be implemented real time. For current scenario, the entire denoising-background removal and clustering of a single image takes around 10 seconds which could be dropped to around milliseconds after implementation of a supervised algorithm for the classification process.

8. Conclusion

We presented a novel improved U-Net architecture capable of denoising and background removal within an average of 3ms and also a model developed using SCAN that can cluster wear particles present in an OLVF image into their respective clusters based on their several properties as learnt by the network. This is advantageous for further researches on wear particles and developing a wear model and also enhance the idea of predictive maintenance for a system as the clustering process allows us to count the number of particles that fell into that particular cluster which is a valuable insight about any system.

In current context, we don't have any method to confidently say that the model has segregated the particle into right cluster but the paper has opened a new process through which human evaluators can judge on its output and create another annotated dataset through which a supervised learning could be established so that things can be done more confidently.

The paper has also opened another dimension where the real time image denoising and background removal along with clustering could be done after we are ready with the annotated dataset for the wear particles for supervised learning. This helps to enhance the abilities of OLVF system so that it can be more powerful and intelligent than any other methods of wear particles detection and analysis.

Acknowledgement

This work was partly supported by the National Science Foundation of China (Grant No. 52175113, No. 51905406), the Scientific Research Program Funded by Shaanxi Provincial Education Department under Program 21JK0693, and the Youth Talent Project supported by Xi'an Association for Science and Technology under Grant 095920211326.

REFERENCES

[1]. R. P. B. T. L. M. V Macián, "Applying analytical ferrography as a technique to detect failures in diesel engine fuel injection systems," *Wear*, pp. 562-566, 2006.

[2]. B. Roylance, "Ferrography—then and now," pp. 857-862, 2005.

[3]. J. W. Y. P. A. Y. Z. Tonghai Wu, "Description of Wear Debris from On-Line Ferrograph Images by Their Statistical Color," *Tribology Transactions*, 2012.

[4]. T. W. S. W. Z. P. Y Peng, "Oxidation wear monitoring based on the color extraction of on-line wear debris," *Wear*, vol. 332, p. 1151–1157, 2015.

[5]. IBM, "IBM Cloud Learn Hub," IBM, 19 08 2020. [Online]. Available: https://www.ibm.com/cloud/learn/supervised-learning. [Accessed 04 2022].

[6]. H. L. Y. Z. Y. e. a. Lu, "Underwater optical image processing: a comprehensive review.," *International Journal for Advanced Robotics System,* vol. 17, no. 5, 30 September 2020.

[7]. I. Q. L. J. J.-P. M. StèphaneBazeille, "Automatic Underwater Image Pre-Processing," *HAL,* pp. 16-19, 2010.

[8]. R. S. a. S. Corchs, "Underwater Image Processing: State of the Art of Restoration and Image Enhancement Methods," *EURASIP Journal on Advances in Signal Processing,* vol. 2010.

[9]. N. A. M. I. Ahmad Shahrizan Abdul Ghani, "Underwater Image Quality Enhancement Through Rayleigh-Stretching and Averaging Image Planes," *Int. J. Nav. Archit. Ocean Eng.,* 2014.

[10]. R. P. O. S. H. a. H. J. Eustice, "Underwater image toolbox for optical image processing and mosaicking in MATLAB," in *Proceedings International Symposium on Underwater Technology*, Tokyo, 2002.

[11]. M. Y. W. a. A. E. Hitam, " Mixture contrast limited adaptive histogram equalization for underwater image enhancement," in *IEEE International Conference on Computer Applications Technology (ICCAT)*, Sousse, 2013.

[12]. K. O. M. J. A. S. R. a. T. A. Iqbal, "Enhancing the low quality images using unsupervised color correction method," in *International Conference on System Man and Cybernatics (SMC)*, Istanbul, 2010.

[13]. P. C. C. Yan-Tsung Peng, "Underwater Image Restoration Based on Image Blurriness and Light Absorption," *IEEE TRANSACTIONS ON IMAGE PROCESSING,* vol. 26, 2017.

[14]. X. Z. a. P. C. C. Y.-T. Peng, "Single Underwater Image enhancement using depth estimation based on blurriness," in *IEEE Int. Conf. Imag. Process. (ICIP)*, 2015.

[15]. P. Soille, "Morphological image analysis," in *Principles and Applications*, Berlin, Germany, Springer-Verlag, 1999, pp. 173-174.

[16]. D. L. a. Y. W. A. Levin, "A closed-form solution to natural image mating," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 30, pp. 228-242, 2008.

[17]. J. S. a. X. T. K. He, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell,* vol. 35, no. 6, pp. 1397–1409,, 2012.

[18]. Y. W. D. H. T. D. Wei Song, "A Rapid Scene Depth Estimation Model Based on Underwater Light Attenuation Prior for Underwater Image Restoration," *IEEE*, 2018.

[19]. P. F. a. T. B. Olaf Ronneberger, "U-Net: Convolutional Networks for Biomedical Image Segmentation," Arxiv, Germany, 2015.

[20]. Keras API Reference, "Keras," Keras, [Online]. Available: https://keras.io/api/layers/pooling_layers/max_pooling2d/#:~:text=Max%20pooling%20operation%20for%202D,by%20strides%20along%20each%20dimension.. [Accessed 2022].

[21]. V. G. Babita Sharma, "Adaptive Approach for Zooming Images Using Bilinear Interpolation Technique," *International Journal of Scientific & Engineering Research,* vol. 6, no. 10, pp. 18-21, 2015.

[22]. L. J. Nazanin Beheshti, *Squeeze U-Net: A Memory and Energy Efficient Image Segmentation Network*, Computer Vision Foundation, 2020.

[23]. J. Brownlee, "Machine Learning Mastery," 29 April 2019. [Online]. Available: https://machinelearningmastery.com/introduction-to-1x1-convolutions-to-reduce-the-complexity-of-convolutional-neural-networks/. [Accessed 20 April 2022].

[24]. S. T, "Analytics Vidhya," 24 August 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/all-you-need-to-know-about-skip-connections/#:~:text=awesome%20concept%20now.-,What%20are%20Skip%20Connections%3F,different%20problems%20in%20different%20architectures.. [Accessed 21 January 2022].

[25]. K. E. A. S. T. G. A. S. Jasper R. R. Uijlings, "Selective Search for Object Recognition," *International Journal of Computer Vision*, pp. 154-171, 2013.

[26]. V. Alto, "Medium," 02 January 2021. [Online]. Available: https://medium.com/dataseries/understanding-selective-search-for-object-detection-3f38709067d7. [Accessed 27 January 2022].

[27]. S. V. S. G. M. P. L. V. G. Wouter Van Gansbeke, "SCAN: Learning to Classify Images without Labels," *arXiv*, vol. 2, pp. 1-26, 2020.

[28]. M. Modi, "Medium," 05 July 2020. [Online]. Available: https://medium.com/visionwizard/unconventional-image-classification-approach-d37900b62079. [Accessed 2022].

[29]. S. K. Haldar, Mineral Exploration, Second ed., Elsevier, 2018, pp. 47-68.

