PCCP

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue



Cite this: *Phys. Chem. Chem. Phys.*, 2023, **25**, 15970

# Accurate identification and measurement of the precipitate area by two-stage deep neural networks in novel chromium-based alloys†

Zeyu Xia, ‡[a] Kan Ma,‡[c] Sibo Cheng, *[b] Thomas Blackburn, [c] Ziling Peng,[d] Kewei Zhu,[e] Weihang Zhang,[b] Dunhui Xiao,[f] Alexander J Knowles[c] and Rossella Arcucci[g]

The performance of advanced materials for extreme environments is underpinned by their microstructure, such as the size and distribution of nano- to micro-sized reinforcing phase(s). Chromium-based superalloys are a recently proposed alternative to conventional face-centred-cubic superalloys for high-temperature applications, *e.g.*, Concentrated Solar Power. Their development requires the determination of precipitate volume fraction and size distribution using Electron Microscopy (EM), as these properties are crucial for the thermal stability and mechanical properties of chromium superalloys. Traditional approaches to EM image processing utilise filtering with a fixed contrast threshold, leads to weak robustness to background noise and poor generalisability to different materials. It also requires an enormous amount of time for manual object measurements on large datasets. Efficient and accurate object detection and segmentation are therefore highly desired to accelerate the development of novel materials like chromium-based superalloys. To address these bottlenecks, based on YOLOv5 and SegFormer structures, this study proposes an end-to-end, two-stage deep learning scheme, DT-SegNet, to perform object detection and segmentation for EM images. The proposed approach can thus benefit from the training efficiency of CNNs at the detection stage (*i.e.*, a small number of training images required) and the accuracy of the ViT at the segmentation stage. Extensive numerical experiments demonstrate that the proposed DT-SegNet significantly outperforms the state-of-the-art segmentation tools offered by Weka and ilastik regarding a large number of metrics, including accuracy, precision, recall and *F*1-score. This model forms a useful tool to aid alloy development microstructure examinations, and offers significant advantages to address the large datasets associated with high-throughput alloy development approaches.

Received 25th January 2023,
Accepted 9th May 2023

DOI: 10.1039/d3cp00402c

rsc.li/pccp



## 1 Introduction

The integration of microstructural and chemical characterization, property evaluation, and numerical tools is essential in modern-day metallurgy to enhance the design, development, and deployment of alloys. This integration is facilitated using the Integrated Computational Materials Engineering ICME frameworks and the Materials Genome Initiative MGI.[1] In computational materials science, learning-based approaches have been incorporated into the CALculation of PHAse Diagram CALPHAD models to enable the high-throughput calculations for *ab initio* modelling, phase boundary identification, and kinetics modelling.[2–4] These approaches not only accelerate the material design in an "infinite" material design space, but are also highly desirable to be paired with high-throughput experimental investigations and subsequent data processing for the analysis of novel materials, including their microstructure recognition on large micrograph image datasets.

In the microstructure of many engineering alloys and novel alloys, secondary phases are known to be influential on mechanical behaviour. The volume fraction, size and shape of secondary phases or particles in alloys are, therefore, important

[a] *Queensland University of Technology, Queensland 4006, Australia*
[b] *Data Science Institute, Department of Computing, Imperial College London, London SW7 2AZ, UK. E-mail: sibo.cheng@imperial.ac.uk*
[c] *School of Metallurgy and Materials, University of Birmingham, Birmingham B15 2SQ, UK*
[d] *Institute of Advanced Science Facilities, Shenzhen 518107, P. R. China*
[e] *Department of Computer Science, University of York, York Y010 5DD, UK*
[f] *School of Mathematical Sciences, Tongji University, Shanghai 200092, P. R. China*
[g] *Department of Earth science & engineering, Imperial College London, London SW7 2BP, UK*

† Electronic supplementary information (ESI) available. See DOI: https://doi.org/10.1039/d3cp00402c

‡ These authors contributed equally to this work.

parameters. Equipped with an optical microscope or, more frequently today, an Electron Microscopy EM, images of microstructure can be easily acquired, and image-driven microstructure analysis is an essential step to obtain the information of second phases or particles. Accurate segmentation is thus of the utmost importance for microstructure recognition. The most used microscructure segmentation method in material science is the manual selection of thresholds, such as using the most popular free software ImageJ,[5] or using an automatic global thresholding algorithm,[6] but it is not suitable for many cases, especially subtle thresholds for multi-modal histogram images in another word, images with varying background contrast such as Transmission Electron Microscopy TEM images mentioned in Amandine's work.[7] Although many computer vision segmentation techniques such as edge detection, region-based segmentation, partial differential equation, and watershed segmentation can improved the accuracy by using more carefully engineered features,[8] they all present limitations in sensitivity to noise, impractical use for a large amount of data.

Today, machine-learning-based segmentation techniques have been widely applied not only to cell tracking,[9] brain tumour segmentation,[10] autonomous driving,[11,12] geographic,[13–15] and also material science.[16] DeCost *et al.*[17] adopted the "bag of visual feature" image representation for Support Vector Machine SVM model to perform microstructure classification. Based on Fully Convolutional Neural Network FCNN, Azimi *et al.*[18] proposed a robust method to classify certain microstructural constituents of low carbon steel for steel quality appreciation. DeCost *et al.*[19] proposed a DCNN-based model to perform segmentation on complex microstructures. Ma *et al.*[20] proposed a local processing method and a symmetric rectification so that their base model, DeepLab, outperforms existing segmentation models. Inspired by U-Net, Roberts *et al.*[21] proposed the CNN-based DefectSegNet to perform crystallographic defects segmentation in structural alloys. Cohn *et al.*[22] proposed an instance segmentation tool for metal powered particles produced from gas atomization based on Mask-RCNN, so that researchers can measure the distribution of particle sizes, as well as measure the satellite content in powder samples. Recently, the segmentation for precipitate analysis using the machine learning tool has been attracting increasing attention. Liu *et al.*[23] proposed a CNN-based model to identify materials descriptors describing $\gamma'$ precipitate coarsening in Co-based superalloys. Wang *et al.*[24] adopted the U-Net segmentation model and a regression model to predict the morphological parameters of the microstructure. Wang *et al.*[25] proposed a framework that consists of a U-Net module and ResNet50 module to detect $\delta$ phase and estimate its area accurately. Softwares integrated with common segmentation models like ilastik pixel classification[1,26] and Weka trainable segmentation[27] have achieved microscopy pixel classification tasks in material science. This emerging topic is attracting increasing attention, and it holds promise for precipitate analysis. Although previous models yielded successful segmentation results, the algorithms used in these models were not state-of-the-art. We propose the implantation of state-of-the-art models like the You Only Look Once YOLO detection model and SegFormer segmentation model, which will allow for higher efficiency and accuracy in segmentation. Efficient and accurate measurement of precipitate size is imperative for the analysis of precipitate size evolution during the ageing heat treatment, which determines their coarsening rate. In addition, the comparison between the previous models and models to date for precipitate analysis has not been addressed.

Given that precipitates have, in general, a regular shape, *e.g.* spherical or cuboidal, a general dataset containing different conditions of microstructures can be created from existing samples of materials to train a deep learning model, which can then intelligently perform the analysis in new datasets. In this context, this work highlights the application of a deep learning method to precipitate detection in the microstructural design of materials for high-temperature applications. High-temperature materials, including face-centred-cubic fcc nickel-based and cobalt-based superalloys, undergo precipitation during heat treatment, leading to precipitate strengthening.[28,29] In these state-of-the-art materials, the precipitate volume fraction and size distribution after different heat treatments are crucial for the strength and creep resistance of such alloys. The coarsening of precipitates in fcc-superalloys have been extensively studied[30–33] and enable the precise control of their microstructure and desired properties. Developing novel materials, such as body-centredcubic bcc chromium-based[34,35] and iron-based ferritic superalloy,[36,37] also requires extensive microstructural observations after various heat treatments using EM and lengthy data processing times. Image processing refers to identifying the matrix and precipitate phases, followed by measuring the size distribution and area fraction of the precipitate.

Cr-superalloys, principally Chromium (Cr)–Nickel–Aluminide (NiAl) alloys consisting of a disordered bcc Cr matrix with an A2 structure strengthened by ordered bcc NiAl intermetallics with a B2 structure, have been identified as potential alternatives to nickel-based superalloys and advanced austenitic steels for high-temperature applications.[34,35,38] Cr-Superalloys with Fe additions have been further developed in the framework of a European project COMPASsCO2 for advanced Concentrated Solar Power applications.[39] Cr offers advantages such as a high melting point, low cost, good oxidation resistance, and low mass density. However, Cr–NiAl alloys are a nascent class of materials, and their precipitate coarsening kinetics are yet to be investigated.

The size of the B2 precipitates and their morphology is important for the mechanical behaviour of these NiAl-strengthened alloys, such as achieving a high yield strength or creep resistance[40,41] in Fe–NiAl ferritic alloy systems. Studying the coarsening rate also contributes to the evaluation of material parameters of new alloys, such as interfacial energy and diffusion coefficients, which will be utilised in physical models for CALPHAD and ICME. However, the precipitate coarsening alongside the structure–property relationship is principally unknown for Cr-superalloys. Moreover, calculating coarsening rates requires the measurement of precipitate size in numerous samples aged at various temperatures and ageing times, which is laborious through traditional methods.

Table 1 Cr-superalloy sample compositions in atomic percent (at%) and their respective heat treatment conditions

| Label | Composition | Heat treatment[a] | Phases expected | SEM observation |
|---|---|---|---|---|
| 5-5 | Cr–5Ni–5Al | H + A1 | A2/B2 | Matrix – precipitates |
| 5-5-10 | Cr–5Ni–5Al–10Fe | H + A2 | A2/B2 | Matrix – precipitates |
| 10-10-20-4 h | Cr–10Ni–10Al–20Fe | H + A1 | A2/B2 | Matrix – precipitates |
| 10-10-20-100 h | Cr–10Ni–10Al–20Fe | H + A3 | A2/B2 | Matrix – precipitates |

[a] Heat treatment annotation. H: homogenisation at 1400 °C for 20 hours. A1: ageing at 1200 °C for 4 hours. A2: ageing at 1000 °C for 100 hours. A3: ageing at 1200 °C for 100 hours.

In this paper, a new, robust, and accurate 2-stage segmentation model on novel $\beta$–$\beta'$ chromium-based alloys (Cr-superalloys for short) is proposed. This work aims to develop a learning-based approach to investigate the precipitate area and size distribution in Cr-superalloys. In summary, this paper aims to highlight the following:

- Manufacture of Cr-superalloys with various heat treatments to produce an A2–B2 microstructure with B2–NiAl sizes varying from nm–μm scales.
- Development of an end-to-end object segmentation model using a two-stage DNN DT-SegNet for object segmentation on EM images with separate training of the detection and segmentation networks.
- Application of the DT-SegNet to determine the area fraction and size distribution of precipitates in Cr-superalloys.
- Demonstration of a developed DT-SegNet can outperform the state-of-the-art segmentation methods in terms of $F1$-score (Table 1).

# 2 Material and methodology

## 2.1 Studied materials

After ageing, B2–NiAl spherical precipitates are observed in the SEM in all samples, as shown in Fig. 1. The size of precipitates varies from nano-scale to micro-scale depending on ageing conditions. The contrast of the precipitates and matrix phases also varies due to the polishing effect on different precipitate sizes. Six SEM images, taken at a suitable magnification to contain tens of precipitates, are captured of each sample and used to train the model. In those images, precipitates with their boundaries were carefully identified and manually labelled for the training of the models, as illustrated in *cf.* Fig. 2. Since most precipitates had a spherical morphology, their sizes were approximately calculated as a function of their radius $r = \sqrt{A/\pi}$ with $A$ being the measured area.

## 2.2 The proposed model: DT-SegNet

Driven by the analysis of previous methods, we proposed a novel end-to-end two-stage deep learning scheme combining a Detection (DT) stage and a Segmentation stage Network, termed as DT-SegNet. As shown in Fig. 2, the network is designed for precipitate identification and measurement in two stages: a detection stage based on YOLOv5[42] and a segmentation stage based on SegFormer.[43]

YOLO model is an end-to-end object-detection model which processes the images in the form of small grid regions.

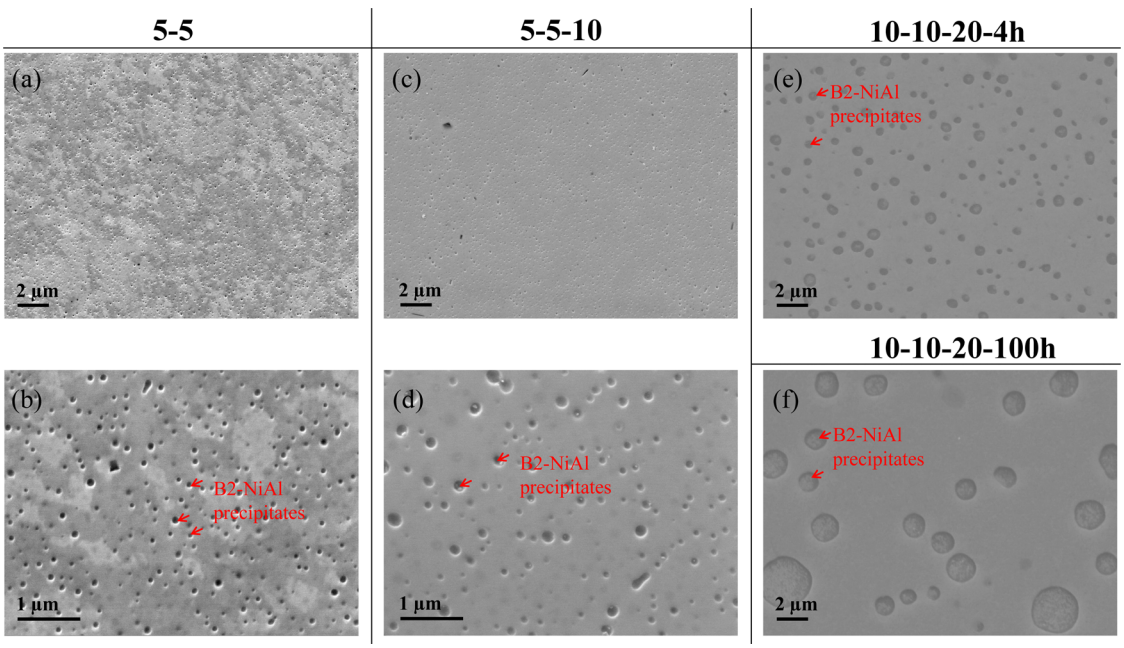


Fig. 1 SEM micrographs showing the general microstructure of (a) Cr–5Ni–5Al, (c) Cr–5Ni–5Al–10Fe, (e) and (f) Cr–10Ni–10Al–20Fe aged differently. (b) and (d) are zoomed images respectively of (a) and (c).

 

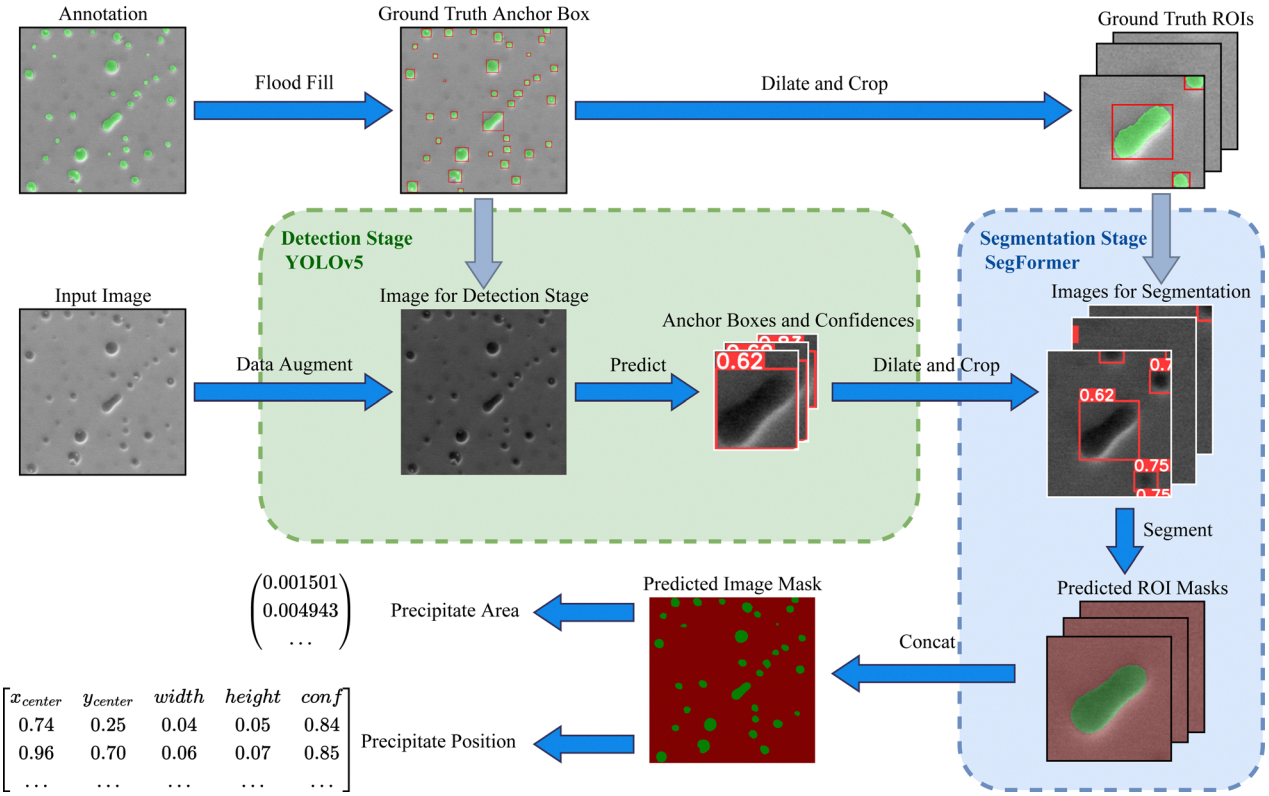


Fig. 2 Architecture and pipeline of the proposed DT-SegNet. The first step is to pre-process the input image to fit the input size of the detection network. Once predicted, the anchor boxes are dilated and cropped before feeding into the segmentation network. Finally, the accurate mask, area and position of precipitate objects are derived.

Calculating the target bounding boxes and confidences based on weights in smaller regions is crucial to accelerating and enhancing detection accuracy. The SegFormer is a segmentation network consisting of a hierarchical Transformer Encoder backbone, an all-MLP decoder neck, and an MLP segmentation head. This design allows effective multi-scale extraction and utilisation of critical features without using complex decoders to improve performance and reduce computational costs.

The first detection stage aims to locate the anchor boxes of precipitates with their confidence. In this stage, the input shape of EM images is resized to 1280 px × 1280 px. Appropriate data augmentations (such as random scaling, random flipping, mosaic and normalisation) are applied to alleviate the lack of generalisation caused by limited training data. After pre-processing and augmentation, the image is delivered to a YOLOv5 network to produce a list of predicted regions with their confidence.

In the second segmentation stage, regions are filtered by a hyper-parameter of the confidence threshold to remove falsely detected regions caused by background noises. To include background information, detected regions are then dilated by 50% of the original size. Once each extended region is cropped, the new region with extra background information is referred to as the Region of Interest ROI, which acts as the input for the SegFormer model. The segmentation model then performs the semantic segmentation task, producing a pixel-wise mask of each precipitate.

Finally, a list of all detected precipitates with their regions, positions and masks can be used to perform precipitate area calculations and other downstream tasks. The overall pipeline is shown in Fig. 2.

**2.2.1 Detection stage.** Traditional region proposal neural networks, like Mask R-CNN[44] and Convolutional Neural Network CNN[45], use bounding boxes and classify detected objects in two stages resulting in a more extensive computation cost and less awareness of global features. Also, as they scan the whole image with a multi-scale sliding window, the number of windows needs to be pre-defined. Unsatisfactory regions may be detected if only a fixed number of window templates are applied. Compared with two-stage methods, the one-stage YOLO model directly uses joint grid regression to predict both the confidence and the bounding box, which is extremely fast and can learn more generic features of the target object.[46]

YOLO is a family of end-to-end networks for object detection. The YOLOv1[47] is the first end-to-end differentiable neural network which combines object classification and object detection. The author of YOLOv3[48] added connections to the backbone network layers, which enables the prediction to be made at three different levels of granularity, resulting in a significant performance gain on small objects. YOLOv4[49] uses new features, including Cross Stage Partial CSP connections, cross mini-batch normalisation, self-adversarial-training, mosaic data augmentation and complete Intersection over Union IoU loss to improve the accuracy and detection speed significantly.

YOLOv5[42] is the first YOLO implementation using the PyTorch framework instead of the Darknet framework. Its novel design includes adaptive anchor boxes, allowing the network to select the most optimal anchor box that fits the dataset. One of the most significant improvements of YOLOv5 is its 6 × 6 Conv2d layer, which reduces the number of parameters without impacting model performance. To increase the inference speed, it also replaces the SPP structure with Spatial Pyramid Pooling SPPF, which is faster with the same output.

An overview of the YOLO model architecture is shown in Fig. 3. YOLOv5 is a CNN-based one-stage object detection network consisting of a backbone of CSP-Darknet53,[50] a neck of SPPF and Path Aggregation Network PANet,[51] and three YOLOv3 heads. As seen in the figure, the backbone extracts influential features from input images, and then the neck aggregates all the captured features. Finally, the locations of the objects are computed by the heads. Three heads calculate bounding boxes and probability maps in the grid system and then use all predictions to calculate the final prediction. In summary, YOLOv5 adopts all these state-of-the-art techniques in its user-friendly code base, resulting in an outstanding performance with fast speed.[42] Its detection functionality and the ability to detect multi-scale objects benefit our task.

YOLOv5 has five models in different scales, all having the same model architecture. The authors designed two parameters: ''depth_multiple'' and ''width_multiple'', to control the model scale by multiplying pre-defined constants by the depth and the number of convolutional kernels. This simple design enables selecting the network scale based on the specific problem scale without changing the overall architecture. In this study, multiple networks are tested. After comparing each network, the backbone based on the pre-trained YOLOv5l model with an input size of 1280 px × 1280 px is selected for the detection stage. A further explanation of the detection model selection is in Section 4.5.

The input of the detection stage is a single-channel 2D image. In order to fit all data onto a standard scale, data augmentation is applied to the dataset. The images are resized to 1280 px × 1280 px to maintain a consistent network input shape. The output of the detection stage is a list of target anchor boxes for each precipitate. Each anchor box, with corresponding confidence, is represented in the YOLO format ($x$-centre, $y$-centre, width, height, and confidence).

In this study, improving the detection performance on the small-scale dataset is essential. YOLOv5 utilises several data augmentations to make the most use of the dataset. By applying a set of data augmentation, it is possible to improve the performance without decreasing inference speed.[49] Excluding common data augmentation strategies like random scaling, cropping, and random arranging, YOLOv5 introduces two more strategies: Mosaic (first introduced in YOLOv4) and Mixup, which significantly improves the detection accuracy of small objects. Following Bochkovskiy's work,[49] four training images are concatenated to allow object detection outside their ordinary context. Batch normalisation[52] is applied on the concatenated image to reduce the need for a large mini-batch size. This strategy helps generalise the target object by learning the most common features of the target object. Mixup[53] is another principle to enhance training performance. By generating convex combinations of different sample images, it regularises the network to select simple linear behaviours to be robust to adversarial inputs. However, since the information of precipitates lies on their edge and internal-external difference, the mixup operation causes a loss of these essential attributes.

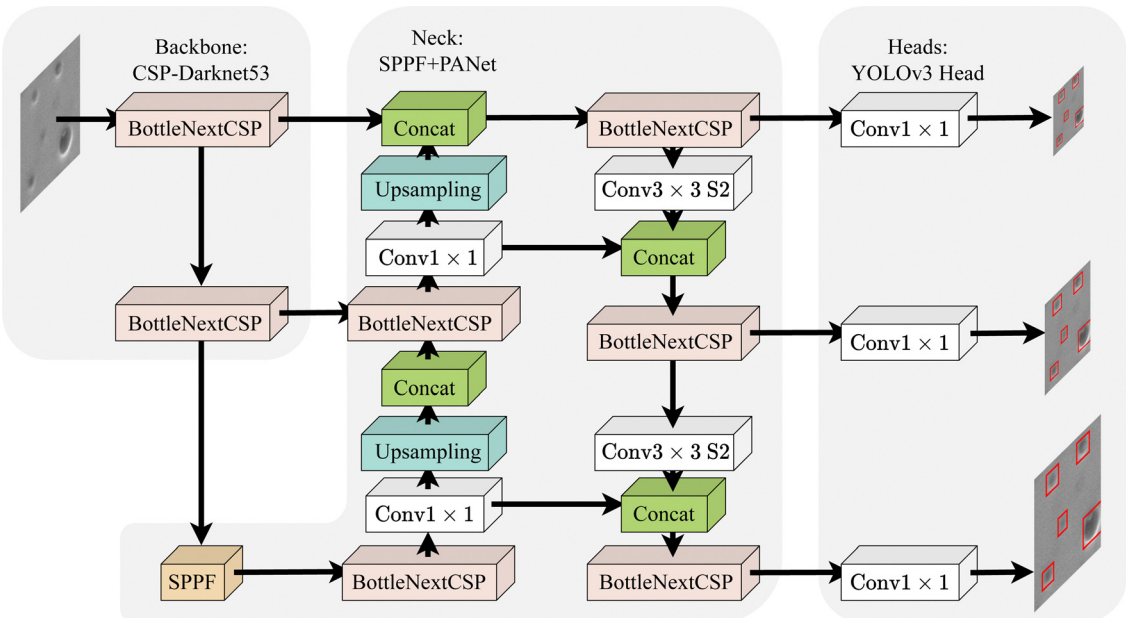


Fig. 3 Illustration of the detection stage YOLO model. The model consists of three parts: backbone, neck and heads. The backbone extracts features, the neck performs feature fusion, and the heads detect the object.

 

Therefore, the mixup operation is excluded from our data augmentation method set.

**2.2.2 Segmentation stage.** As can be seen from Fig. 4, the network for the segmentation stage, SegFormer,[43] consists of an Encoder of four Transformer[54] modules as the backbone, an MLP decoder as the neck and an MLP segmentation head. Four Transformer modules' backbone extracts coarse-grained and fine-grained features. After that, the neck fuses the extracted features and passes them to the segmentation head so that the head can make a final prediction of the semantic segmentation mask. In the proposed DT-SegNet, essential features such as edges and internal textures are captured and generalised, enabling more precise pixel classification and segmentation on edges with good noise resistance.

Research on ViT[55] has suggested that a Transformer directly applied to images performs significantly better than traditional CNN networks. However, the columnar structure of such a model makes it computationally expensive. Additionally, ViT only outputs feature maps of a fixed resolution, which can cause inaccuracy in the segmentation task. To solve these problems, SegFormer[43] proposed a simple and efficient design that unifies the Transformer module with lightweight MLP decoders. This design achieves excellent performance gains while maintaining a reasonable computation cost.

Although the shape, internal texture, and edge brightness between precipitates, most can be detected by their edges. Therefore, fully extracting the edge and perceiving more background information can help distinguish edges from the background. Thus, image dilation is designed ahead of the segmentation stage. In this operation, the boundary of each target anchor box is expanded twice in both weight and height, then resized to 512 px × 512 px. The necessary edge information can be kept by applying dilation, making the segmentation stage less sensitive to false precipitate detection. The extra background information also helps the segmentation network to have more information about the context of the target object. The dilated region with extra background information is named ROI in this paper.

## 3 Dataset

This article conducts experiments on the dataset generated in this study, which contains $N = 24$ SEM two-dimensional images. Details of the dataset are shown in Table 2. The data is split into training, validation and test sets (in a 6 : 2 : 2 ratio) using the hold-out method to ensure even distribution in each set. Due to the small scale of the dataset, images with similar image features were manually assigned into different sets. By doing this, a comparison of the robustness of different models can be made. The result of evaluating the precipitate areas in both the original image and the ROI shows that the dilation operation significantly improves the precipitate area percentage.

The bar charts in Fig. 5 show the distribution of precipitate scales in three datasets. It can be observed that in all the datasets, most of the precipitate area percentage is under

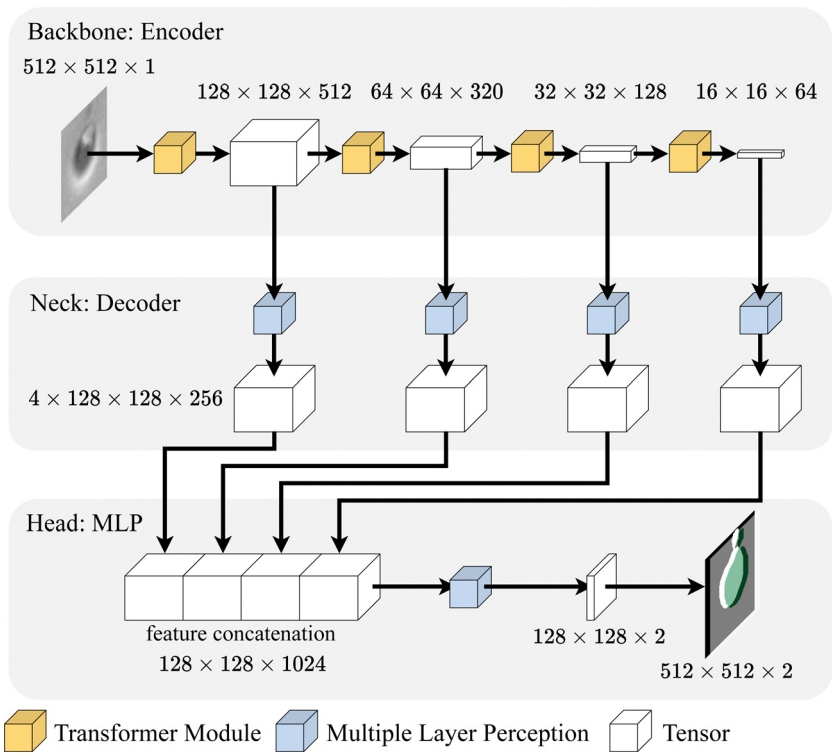


Fig. 4 Illustration of the segmentation stage SegFormer model. This model consists of four transformer modules as an encoder backbone, an all-MLP module as the decoder neck and an MLP module as the head.

Table 2 Statistics of the three datasets split by the ratio of 6 : 2 : 2. The precipitate area ratio in ROI was significantly higher than in the raw input image

| | | Precipitates | | |
|---|---|---|---|---|
| Dataset | Image count | Count | % in Image | % in ROI |
| Training | 15 | 1674 | 9.69 | 23.21 |
| Validation | 4 | 355 | 6.34 | 21.86 |
| Test | 5 | 243 | 9.73 | 25.64 |

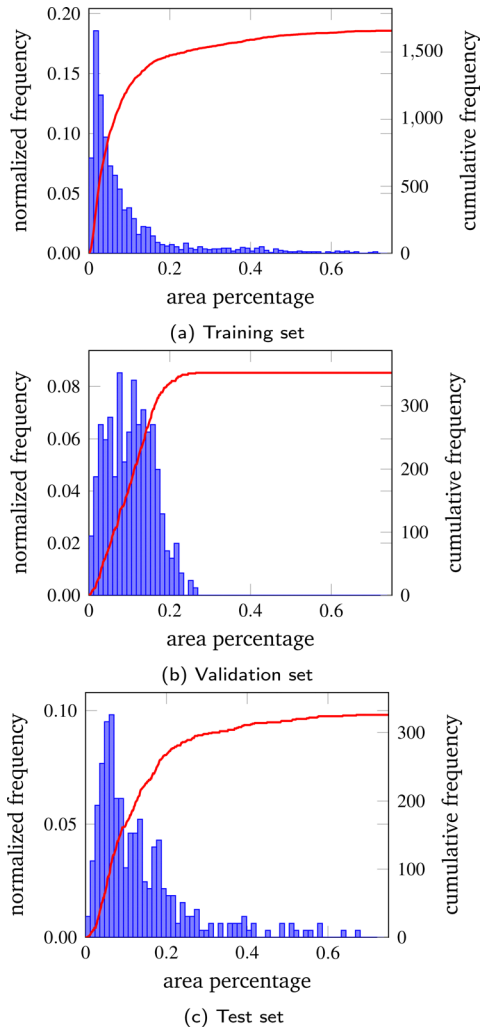


Fig. 5 Distribution of precipitate scales in the three sets. The bar shows the normalised frequency on each dataset, and the curve shows the cumulative frequency.

0.2%. However, the training set has few aberrant precipitates with relative scales larger than 0.2%. As for the validation set, the distribution shows a narrower overall range of 0.3%. The test set contains a set of images where most of the precipitate scales are below 0.2%, whereas some irregular samples with large scales exist.

All three datasets have the most precipitates with areas under 0.2% of the total area.

A three-phase process is followed to produce ground truth for this dataset. Initially, images are labelled interactively using PaddleSeg,[56] and then manually refined using Adobe Photoshop. The shapes and boundaries are corrected during this process. Once finished, the segmentation labels are converted into YOLO-format anchor boxes using the flood-filling algorithm. The final stage comprises a precipitate region correcting step using LabelImg.[57] In this process, overlapping anchor boxes are separated into individual anchor boxes.

# 4 Results and discussion

To thoroughly evaluate the performance of DT-SegNet, this article first experiments with multiple sets of settings on both the YOLOv5[42] network and the SegFormer[43] network to find the most optimised backbone configuration. Then, this article selects five representative methods implemented in two software and four state-of-the-art CNN models in the field of general image segmentation as a comparative experiment. Lastly, this article performs a visualisation analysis on four test images to explain the outcome of each method.

## 4.1 Implementation details

This model is implemented based on the official PyTorch YOLOv5 v6.1 implementation[42] and PaddleSeg v2.7[56] using the PaddlePaddle framework.

At the detection stage, auto-detection of the batch size is used. Minimum epochs of 300 are performed with an early-stopping regularisation of 150-epoch patience. The checkpoint is kept at each epoch. A compound cost function of objectness score, class probability score, and bounding box regression score, a Stochastic Gradient Descent SGD optimiser of 0.01 learning rate and a learning rate scheduler of LambdaLR are used. At this stage, data augmentation of mosaic, copy-paste, random scaling, flipping, hue, saturation adjustment, and normalisation processes are used. Due to the limitation in the dataset scale, the official pre-trained model on the Common Objects in Context COCO 2017 dataset[58] is used for the model to learn more general object features. This dataset includes 80 classes of images with labels such as human, bicycle, traffic light, bird, food, and book.

At the segmentation stage, a batch size of 1, a maximum of 80 000 training epochs, and a checkpoint save interval of 200 are used. CrossEntropyLoss cost function, the AdamW optimiser ($\beta_1$ = 0.9, $\beta_2$ = 0.999, weight decay = 0.01) and a PolynomialDecay learning rate scheduler with learning rate 0.00006 are adopted in our experiments. All images are normalised and applied with random horizontal and vertical flips at this stage. Pretrained MixVisionTransformer models on ImageNet-1K dataset[59] are used.

Other hyper-parameters from both models are maintained as default in their original implementation. The model with the best performance on the validation set is selected as the best model.

The ''varying contrast'' means the difference between foreground and background pixels varies. Traditional methods that apply a constant threshold or a cross-correlation with a Gaussian window[60] provided by the OpenCV library need to handle this problem better. In our work, we used normalisation in the data pre-processing pipeline to maximise the margin of different classes of pixels. Then, the encoder module in our network can perform detection and segmentation tasks from images with different contrasts.

### 4.2 Baseline approaches

In this study, several widely utilised machine learning methodologies, namely, Fast Random Forest FRF and MLP in the Weka software,[27] and Linear Discriminant Analysis LDA, RF, and MLP in ilastik software,[1] are deemed as foundational models for comparison purposes. This study also includes a comparative analysis of contemporary state-of-the-art end-to-end deep learning networks, including U-Net, UNet 3+, DeepLabV3+ and SegFormer. The proposed DT-SegNet scheme is compared against these methods using the same training and test datasets.

RF[61] is a decision-tree-based learning method. It works by building an ensemble of decision trees based on input features. During prediction, the model combines the prediction from all trees to make a final prediction, resulting in a better generalization outcome than a single decision tree. FRF[61] is similar to the standard RF algorithm, but with some modifications to accelerate its speed and reduce memory usage. Based on Java and implemented in Trainable Weka Segmentation,[27] it uses a sub-sampling technique to randomly select a subset of the features and instances for each tree in the forest. It also uses a heuristic algorithm to select the best splitting point at each node, which further improves the model speed. MLP[62] is a type of neural network composed of multiple layers of fully-connected artificial neurons. It uses a back-propagation algorithm to adjust the weights of each neuron based on the error between model prediction and ground truth. LDA[63] is a statistical technique that finds a linear combination of input features that maximizes the separation between different classes. It models the distribution of input features in each class and uses the between-class variance to the within-class variance ratio to calculate the optimal discriminant space for classifying new image pixels. Support Vector Machines C-Support SVC[64] is a soft-margin classification algorithm using a regularisation parameter of C to control the balance between maximizing the margin and minimizing the classification error. U-Net[65] is a widely used CNN model initially designed to solve biomedical image segmentation challenges. It consists of a contraction path, an expansion path, and skip connections that allow the expanding path to use information from the contracting path. This enables it to achieve high accuracy and preserve the original spatial resolution. UNet 3+[66] is an extension of the previous U-Net and its variants. By adding more encoder and decoder layers and introducing dense skip connections and deep supervisions, it has achieved state-of-the-art performance on several medical image segmentation benchmarks. DeepLabV3+[67] is a CNN model that uses a modified atrous spatial pyramid pooling module to capture contextual information over multiple scales and uses a decoder module to produce pixel-wise predictions. SegFormer[43] is a CNN architecture segmentation model that uses a Transformer-based Encoder and a Decoder module with multi-scale feature fusion and progressive upsampling.

Weka trainable segmentation[27] is a machine-learning tool for microscopy pixel classification. This study evaluates the segmentation models of FRF and MLP on this software. Weka trainable segmentation version 3.3.2 with Fiji ImageJ 1.53t is used. We use the default set of standard deviation $\sigma$ in the Gaussian filter applied during the image pre-processing step in all Weka experiments, which are 1.0, 2.0, 4.0, 8.0, and 16.00. Gaussian blur (5 convolutions with 5 variations of $\sigma$), Sobel filter, Hessian, the difference between Gaussians (combination of all $\sigma$), and membrane projections (kernel size of $19 \times 19$) are selected as classification features. In this experiment, the FRF parameter of unlimited max depth, two-decimal-place precision for model output, and two attributes in the random selection is used to generate 200 trees. In this study, the MLP parameter settings of a batch size of 10 000, disabled decay, a learning rate of 0.3, momentum of 0.2, two decimal places, and a validation stage set the size of 20 with a threshold of 20. Both methods are trained with balance classes enabled, which filter more populated foreground pixel samples and duplicates less numerous background pixel samples.

Ilastik pixel classification[1] is an interactive machine-learning tool for bio-image analysis. Segmentation models LDA, RF and SVC are experimented for comparison. In this study, ilastik version 1.4.0rc6 is used. As ilastik does not provide an interface to tune parameters, all parameters are set as the default value. In the scikit-learn implementation, the default margin parameter C for SVC is 1.0, with an RBF kernel and probability estimates enabled. It trains features of Color and Intensity (Gaussian Smoothing), Edge (Laplacian of Gaussian, Gaussian Gradient Magnitude, and Difference of Gaussians), and Texture (Structure Tensor Eigenvalues and Hessian of Gaussian Eigenvalues) for all images using a $\sigma$ of 0.30, 0.70, 1.00, 1.60, 3.50, 5.00 and 10.00. All the methods are implemented on the scikit-learn backend.

Four single-stage segmentation models are trained and inferred using PaddleSeg v2.7[56] on the PaddlePaddle framework, with a checkpoint save interval of 100. U-Net is trained with a batch size of 4, a maximum of 40 000 training epochs, no pre-trained model and deconvolution disabled. UNet 3+ is trained with a batch size of 2, a maximum of 40 000 training epochs, no pre-trained model, batch normalisation enabled, classification-guided module disabled, and deep supervision disabled. DeepLabV3+ is trained with a batch size of 2, a maximum of 80 000 training epochs, ImageNet-1K[59] pre-trained ResNet50_vd backbone, a dilation rate of (1, 12, 24, 36), and no pre-trained model. SegFormer B0 and B1 are trained with a batch size of 1 and a maximum of 80 000 training epochs. CrossEntropyLoss cost function, the AdamW optimiser ($\beta_1$ = 0.9, $\beta_2$ = 0.999, weight decay = 0.01) and a

PolynomialDecay learning rate scheduler with learning rate 0.00006 are adopted in the experiments for SegFormer. All other models except SegFormer are trained with CrossEntropyLoss cost function, a stochastic gradient descent optimiser (momentum = 0.9, weight decay = 0.00004) and a PolynomialDecay learning rate scheduler with learning rate = 0.01, end_lr = 0 and power = 0.9.

### 4.3 Training environment

All models are trained and inferred on a server with AMD EPYC 7543 CPU, an NVIDIA RTX A5000 graphics card and 32 GB Memory. Experiments are under Ubuntu 20.04 operation system, with the programming language Python 3.8, GPU acceleration kit CUDA 11.6, machine learning framework PyTorch 1.13.1 and PaddlePaddle 2.4. Two baseline methods, Weka and ilastik, are trained on a desktop machine running on Windows 10 version 22H2 with an Intel Core i5-9600KF CPU, NVIDIA Geforce GTX 1080 GPU and 32 GB Memory. Due to the online training nature of Weka trainable segmentation and ilastik pixel classification, directly using pixel-wise annotation exhausts system resources and results in the system not responding. Two discrete reasons emerge from this. First, the software generates computationally-heavy features on extensive pixels at their pre-processing stage. Second, there is limited support for GPU acceleration. Therefore, all images in this dataset are relabeled using built-in tools inside both software to solve this problem. As this action may result in a drop in labelling accuracy, the relabeling is repeated twice until all precipitates in the training set are segmented correctly. Another aspect worth noticing is the size of the output model. The trained model of DT-SegNet has a size of 198 MB, compared with 257 MB of the LDA model, 256 MB of the RF model, and 359 MB of the SVC model. However, due to the default unlimited max depth, the FRF model has a size of 1.19 GB. This can make it challenging to deploy such a big model on machines with less memory and CPU power.

### 4.4 Metrics

In this study, a robust comparison of the proposed DT-SegNet against the state-of-the-art tools Weka and ilastik is performed using a wide range of detection and segmentation metrics. Manually labelled data are used as ground truth. The algorithm performances of both detection and segmentation stages are evaluated on the test dataset. Precision, recall, and mAP are measured for the detection stage. TP = Truepositive, TN = Truenegative, FP = Falsepositive, and FN = Falsenegative are denoted.

In the detection stage, two bounding boxes: the prediction box $P$ and the ground truth box $T$ are first defined. Then IoU can be defined as:

$$\mathrm{IoU} = \frac{|P \cap T|}{|P \cap T|}. \tag{1}$$

Based on the IoU, the predicted bounding boxes from the detection model can be classified as TP if the IoU exceeds the IoU threshold (0.6 as default).

Precision is a metric that measures how accurate the prediction is. It is calculated as follows:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{2}$$

Recall demonstrates the ability to find all precipitates, *i.e.*,

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{3}$$

Since precision or recall alone can not fully characterise the prediction effect of the model, a metric that measures the precision and recall jointly is needed. AP[68] is defined as the area under the PRC. The formula is defined as follows:

$$AP = \int_0^1 p(r)\mathrm{d}r \tag{4}$$

where $r$ denotes the recall and $p(r)$ denotes the precision in the function of $r$.

However, the result of AP is heavily affected by the selection of the IoU threshold. The mAP metric[58] is used to alleviate this problem. This metric calculates the average AP score on different IoU thresholds. In this task, $\mathrm{mAP}_{0.5}$ is the AP with the IoU threshold of 0.5. $\mathrm{mAP}_{0.5:0.95}$ computes average AP using IoU thresholds of [0.5, 0.55, 0.60,. . ., 0.95]. Since $\mathrm{mAP}_{0.5:0.95}$ reflects the model performance under most of the IoU thresholds, it is used as the primary metric in the detection stage of this study.

Accuracy, precision, recall, IoU, SSIM, and $F1$-score are evaluated in the segmentation stage. At this stage, the TP predictions as pixels predicted are defined to have the same label as the ground truth annotation.

The pixel-wise accuracy for the segmentation stage is defined as:

$$\mathrm{Pixelaccuracy} = \frac{\mathrm{TN} + \mathrm{TP}}{\mathrm{TN} + \mathrm{FP} + \mathrm{TP} + \mathrm{FN}} \tag{5}$$

This metric represents the number of correctly segmented pixels over the total number of pixels. The area accuracy is also computed, which is defined as follows:

$$\mathrm{Area\ accuracy} = \frac{\text{total precipitate area in prediction}}{\text{total precipitate area in ground truth}} \tag{6}$$

This metric conveys the difference between the predicted and actual area. Precision and recall have the exact definition in the detection stage, but the calculation is performed pixel-wise. The mean IoU is the average IoU on precipitate and background class.

The following formula is used to calculate IoU in the segmentation stage:

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{7}$$

SSIM[69] is used to measure the similarity between the prediction and the ground truth of the exact shape of the precipitate.

 

Table 3 Detection backbone performance on the test set with different settings

| Backbone | Pre-trained | Batch size | Epoch | Input size | Precision | Recall | $mAP_{0.5:0.95}$ | $mAP_{0.5}$ |
|---|---|---|---|---|---|---|---|---|
| YOLOv5n | | 201 | 494 | 640 × 640 | 96.0 | 94.2 | 58.6 | 97.3 |
| YOLOv5s | | 110 | 510 | 640 × 640 | 95.6 | 93.9 | 57.4 | 96.6 |
| YOLOv5m | | 64 | 330 | 640 × 640 | 95.6 | 94.3 | 57.0 | 97.0 |
| YOLOv5l | | 37 | 645 | 640 × 640 | 97.4 | 94.8 | 57.4 | 97.5 |
| YOLOv5n | | 50 | 370 | 1280 × 1280 | 94.7 | 92.9 | 56.0 | 96.2 |
| YOLOv5s | | 29 | 426 | 1280 × 1280 | **97.5** | 93.3 | 59.8 | 97.4 |
| YOLOv5m | | 15 | 486 | 1280 × 1280 | 95.2 | 92.0 | 58.1 | 96.9 |
| YOLOv5l | | 9 | 424 | 1280 × 1280 | 94.0 | **96.3** | 61.0 | 98.2 |
| YOLOv5n | ✓ | 50 | 366 | 1280 × 1280 | 95.4 | 95.9 | 61.5 | 98.1 |
| YOLOv5s | ✓ | 29 | 614 | 1280 × 1280 | 95.9 | 94.5 | 60.6 | 97.7 |
| YOLOv5m | ✓ | 15 | 230 | 1280 × 1280 | 91.6 | 93.9 | 52.9 | 97.5 |
| YOLOv5l | ✓ | 4 | 400 | 1280 × 1280 | 97.4 | 95.7 | **62.5** | **99.0** |

The $F1$-score, defined as

$$F1 = \frac{2 \times \mathrm{TP}}{2 \times \mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \quad (8)$$

can evaluate both precision and recall. Thus, it is selected as the primary metric for comparing model performances in the segmentation stage.

In summary, precision and recall are general metrics for both the detection and segmentation stages. The mAP is used for the detection stage only. Accuracy, IoU, SSIM and $F1$-score are used for the segmentation stage.

### 4.5 Detection backbone

Multiple combinations of models with two input shapes are experimented to find the best configuration in the detection stage. The effectiveness of transfer learning is also explored by using models pretrained at COCO dataset.[58] All models are trained with patience of 150 epochs. Table 3 shows the performance of different networks with different configurations on the test set. According to Ultralytics, the COCO trains natively on 640 px, and benefits can be obtained from increasing the input image size if large amount of small objects exist in the dataset. The results show that increasing input image size from 640 px to 1280 px without pre-training may slightly reduce the model performance on small models such as YOLOv5n but increase the performance gain on large models such as YOLOv5s, YOLOv5m and YOLOv5l. And with pre-training, a faster convergence speed with higher performance is discovered, and models perform better in most settings. According to Luo *et al.*'s work,[70] the effective receptive field increases when more convolutional layers are added, more pooling layers are placed, or convolution stride is higher. In our cases, YOLO networks with an input size of 1280 px × 1280 px have extra convolutional layers than networks with input size 640 px × 640 px. The increased parameters can increase the effective receptive field, thus provides large models with a better generalisation ability on high-resolution input images. The utilisation of pre-trained models shows performance improvement with a 0.6% increase in $mAP_{0.5:0.95}$ on average. The initial weights in pretrained model may accelerate the gradient decent process in a right direction, thus can provide better generalisation ability. After comparison, pre-trained YOLOv5l with an input size of 1280 px × 1280 px is selected for the detection stage.

The $F1$-confidence curve of YOLOv5 is shown in Fig. 6. A higher $F1$-score indicates better detection performance. As seen from the figure, the $F1$-score reaches its peak at 0.97 with a confidence of 0.475. Furthermore, a wide range of confidence thresholds from 0.1 to 0.6 can be selected to perform precipitate detection.

### 4.6 Segmentation backbone

For model selection in the segmentation stage, SegFormer B0 and SegFormer B1 have experimented with an input image size of 512 px. Table 4 shows the performance of different SegFormer networks on the test set. The result shows that excellent performance is achieved using both models. There is a slight improvement regarding the $F1$-score in SegFormer B1, which may be attributed to additional parameters in the Encoder. Because both networks achieve outstanding performance on the task, and the computation cost on SegFormer B1

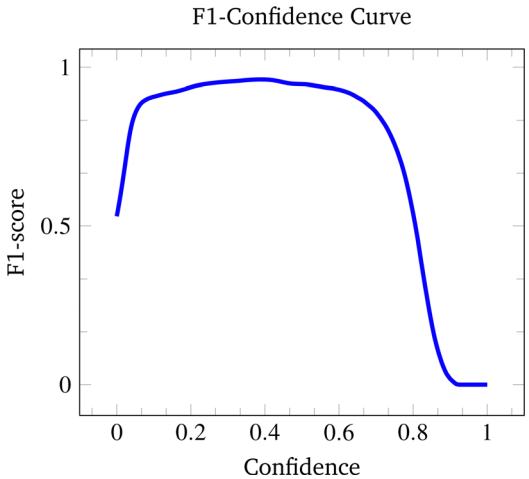


Fig. 6 $F1$-Confidence Curve of pre-trained YOLOv5l with an input size of 1280 px × 1280 px on the validation set. Each point on the line indicates the $F1$-score at the given confidence filter constant. The peak of the $F1$-score is 0.97, reached at the confidence of 0.475.

Table 4 Segmentation backbone performance on the test set with different scales. P: precipitate, B: background

| | | Accuracy | | Precision | | Recall | | IoU | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Backbone | Epoch | Pixel | Area | P | B | P | B | P | B | Mean | SSIM | *F*1 |
| SegFormer B0 | 56 000 | 94.3 | **94.4** | 85.6 | **97.7** | **93.5** | 94.6 | 80.8 | 92.5 | 86.7 | **68.5** | 92.7 |
| SegFormer B1 | 37 000 | **94.5** | 92.3 | **86.9** | 97.3 | 92.3 | **95.2** | **81.0** | **92.7** | **86.9** | 65.6 | **92.9** |

is affordable, SegFormer B1 is chosen as the model in the segmentation stage.

### 4.7 Method comparison

Table 5 shows the segmentation metrics of DT-SegNet against the state-of-the-art machine-learning methods. It can be clearly observed that the proposed DT-SegNet achieves the highest scores among all methods. Compared to the best model in two softwares, Weka running FRF, a significant advantage of DT-SegNet in terms of accuracy (4.2%), precision (6.2%), recall (23.0%) and *F*1-score (18.2%) can be observed. Furthermore, the proposed DT-SegNet exhibits a lower standard deviation on all metrics, showing substantial robustness. Compared to the best CNN model, SegFormer B0, a 2.3% improvement in the *F*1 score can be observed. The standard deviation of the proposed DT-SegNet is also lower. It is worth noticing that although CNN models have achieved outstanding accuracy, they are weak in recall, which means more precipitates are missing. Low recall and high standard deviation may suggest that these methods lack the required robustness to handle the variety of different EM images.

It is also worth mentioning that statistic-based models like LDA can detect most precipitates, resulting in high accuracy and recall. However, this approach induces more false-positive detections, leading to low precision and *F*1-score. Classical machine-learning-based models such as RF, FRF, and MLP, however, have higher IoU and accuracy but miss more precipitates.

To reduce the human bias in the manual dataset split process, as well as make the performance of the proposed DT-SegNet convincing, *K*-fold cross-validation with five folds performed. As shown in Fig. 6, the proposed model performs consistently on different dataset splits. In split 2, the test case has a completely different distribution to the training set, resulting in a slightly lower performance than other splits. Split 5, however, have a balanced distribution in two datasets, resulting in higher performance than other splits (Table 6).

### 4.8 Visual inspection

Segmentation quality can be most intuitively assessed by visualisation, as seen in Fig. 7–10 of four SEM images selected from the test set with outputs from different models. Conditions of the selected images are included in the training dataset.

The original input is shown in the first row, along with the ground truth annotation placed at the right of the first row. The output of the detection stage of DT-SegNet is also shown in the first row. The second row shows the models' predicted output; in this context, green represents the mask of the predicted precipitate. The background pixels are left as it is. For DT-SegNet, the best confidence threshold based on the performance of the validation set, and other methods have their confidence threshold is used as the default value. Perfect segmentation covers all the noticeable precipitates with the best-fitting shape. In the third row, a colourised illustration of taxonomy for segmented pixels is presented: false positive and negative predictions are marked in red. The fourth row shows the predicted precipitate area as a percentage of the original image. In the fifth row, the prediction error is given as a proportion of the input image.

Fig. 7 shows a case with tremendous blurring and background noises frequently encountered in SEM observations. Most methods except LDA successfully detect all precipitates

Table 5 Pixel-wise segmentation performance on our dataset is shown. All results are generated using their best workflows. Pixel classification metrics are used to make comparisons between multiple methods

| Method | Accuracy | Precision | Recall | IoU | SSIM | *F*1 |
|---|---|---|---|---|---|---|
| ilastik (LDA[63]) | 86.8 ± 5.5 | 40.0 ± 17.5 | 82.7 ± 7.4 | 35.9 ± 13.8 | 65.7 ± 18.9 | 51.6 ± 16.2 |
| ilastik (RF[61]) | 93.9 ± 2.9 | 63.7 ± 19.8 | 72.7 ± 20.3 | 49.5 ± 14.5 | 82.4 ± 4.9 | 65.2 ± 13.3 |
| ilastik (SVC[64]) | 75.5 ± 36.1 | 49.4 ± 34.0 | 56.2 ± 40.6 | 23.5 ± 18.4 | 68.8 ± 33.8 | 35.3 ± 23.1 |
| Weka (FRF[61]) | 93.5 ± 6.3 | 76.1 ± 24.2 | 69.6 ± 17.0 | 51.6 ± 9.1 | 82.7 ± 10.3 | 68.0 ± 8.4 |
| Weka (MLP[62]) | 92.0 ± 6.1 | 58.7 ± 25.1 | 79.5 ± 11.8 | 48.6 ± 17.5 | 70.9 ± 17.1 | 64.0 ± 15.4 |
| U-Net[65] | 96.4 ± 4.2 | 80.5 ± 14.9 | 73.8 ± 40.7 | 63.9 ± 35.5 | 92.0 ± 3.5 | 71.3 ± 38.7 |
| UNet 3+[66] | 96.3 ± 4.1 | 87.6 ± 9.1 | 70.4 ± 38.9 | 61.9 ± 34.5 | 91.5 ± 3.5 | 70.0 ± 38.2 |
| DeepLabV3+[67] | 97.6 ± 1.2 | 85.6 ± 9.0 | 87.1 ± 6.1 | 75.5 ± 5.2 | 92.4 ± 1.8 | 85.9 ± 3.4 |
| SegFormer B0[43] | 98.1 ± 0.6 | 87.6 ± 10.4 | 88.9 ± 6.2 | 78.3 ± 6.8 | 93.3 ± 1.1 | 87.7 ± 4.3 |
| SegFormer B1[43] | 97.7 ± 1.0 | 84.1 ± 12.0 | 91.9 ± 7.5 | 77.1 ± 5.3 | 92.4 ± 1.9 | 87.0 ± 3.4 |
| DT-SegNet | 98.3 ± 0.8 | 87.8 ± 8.2 | 92.8 ± 3.4 | 81.9 ± 5.6 | 94.0 ± 1.4 | 90.0 ± 3.3 |

 

Table 6 Pixel-wise segmentation performance of proposed DT-SegNet in *K*-fold cross-validation

| Split | Accuracy | Precision | Recall | IoU | SSIM | *F*1 |
|---|---|---|---|---|---|---|
| 1 | 97.1 ± 2.3 | 90.5 ± 5.6 | 84.0 ± 13.7 | 76.3 ± 9.4 | 89.8 ± 8.6 | 86.3 ± 6.4 |
| 2 | 96.5 ± 2.0 | 79.1 ± 10.7 | 91.5 ± 1.8 | 73.6 ± 9.1 | 88.8 ± 7.2 | 84.5 ± 6.0 |
| 3 | 97.8 ± 0.6 | 86.8 ± 8.1 | 84.0 ± 11.4 | 73.4 ± 5.4 | 93.4 ± 1.2 | 84.6 ± 3.5 |
| 4 | 97.6 ± 0.9 | 83.9 ± 8.1 | 87.2 ± 4.5 | 75.0 ± 8.9 | 91.3 ± 4.7 | 85.5 ± 5.7 |
| 5 | 97.9 ± 2.3 | 90.7 ± 4.6 | 84.5 ± 11.7 | 77.9 ± 11.5 | 93.5 ± 5.7 | 87.2 ± 7.4 |
| Avg | 97.4 ± 1.7 | 86.0 ± 8.4 | 86.3 ± 9.3 | 75.1 ± 8.3 | 91.3 ± 5.8 | 85.5 ± 5.4 |

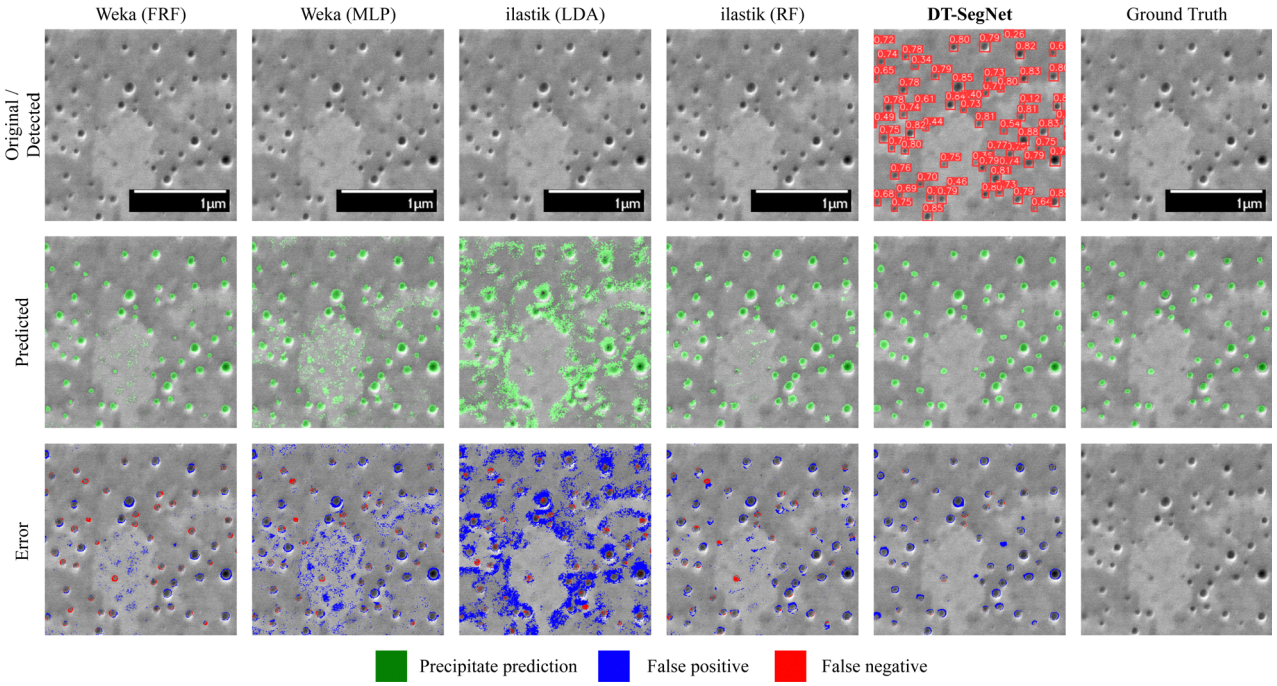


Fig. 7 Visualisation of segmentation results on 5-5 produced by four competing methods and our methods, along with the ground truth annotation.

and segments in good shape, with an error rate lower than 9%. However, the other three baseline models have many false-positive predictions on the white background. It is worth mentioning that there is a spurious precipitate that most of the methods failed to ignore. The false-positive detection may be attributed to its darkness, which shows the real-world experiments' complexity. As a result, they are higher in error ratio compared with DT-SegNet. Although DT-SegNet detects a few background noises as precipitate, most are detected in low confidence and then filtered at the detection stage. Consequently, the segmentation stage only receives the ROI as input, making the model more robust to the uncertain background.

Fig. 8 is a common case of a SESEM image showing nano-scale precipitates. The contrast inside precipitates is different from the contrast of the matrix. Due to the polishing, precipitates are polished slightly more than the matrix. It causes different heights on the precipitate area, which were clearly resolved using SE imaging. Apart from the precipitates exposed on the surface, weak blurry contrast from some embedded precipitates is observed, which are excluded in the observation. It can be seen in the original images that the precipitates have white edges, which can be a helpful feature for models. Decision-tree-based algorithms like FRF and RF can detect most precipitates correctly and have the closest to the ground truth value, with errors near the edge. The error may be attributed to its lack of generalisation of objects in an irregular shape. LDA fails to differentiate the edges of precipitates, so the detected area tends to be considerably larger than the ground truth. The MLP produces a more robust result, but due to its small model size, the model has difficulties distinguishing the background noise from precipitates. DT-SegNet has perfect detection results on the input image (lowest error ratio), showing the model is robust to the background noises. However, it is still challenging for the model to fully detect small-scale precipitates, and the segmentation task of abnormal precipitates may still be inaccurate.

Fig. 9 shows a case of the SESEM image with nano-scale precipitates. In this figure, precipitates are larger than those in Fig. 8, and the contrast is different. The edge is apparent, but some light points exist in these large precipitates. In this scenario, all models can better detect the precipitate area. However, both Weka- and ilastik-based methods fail to segment

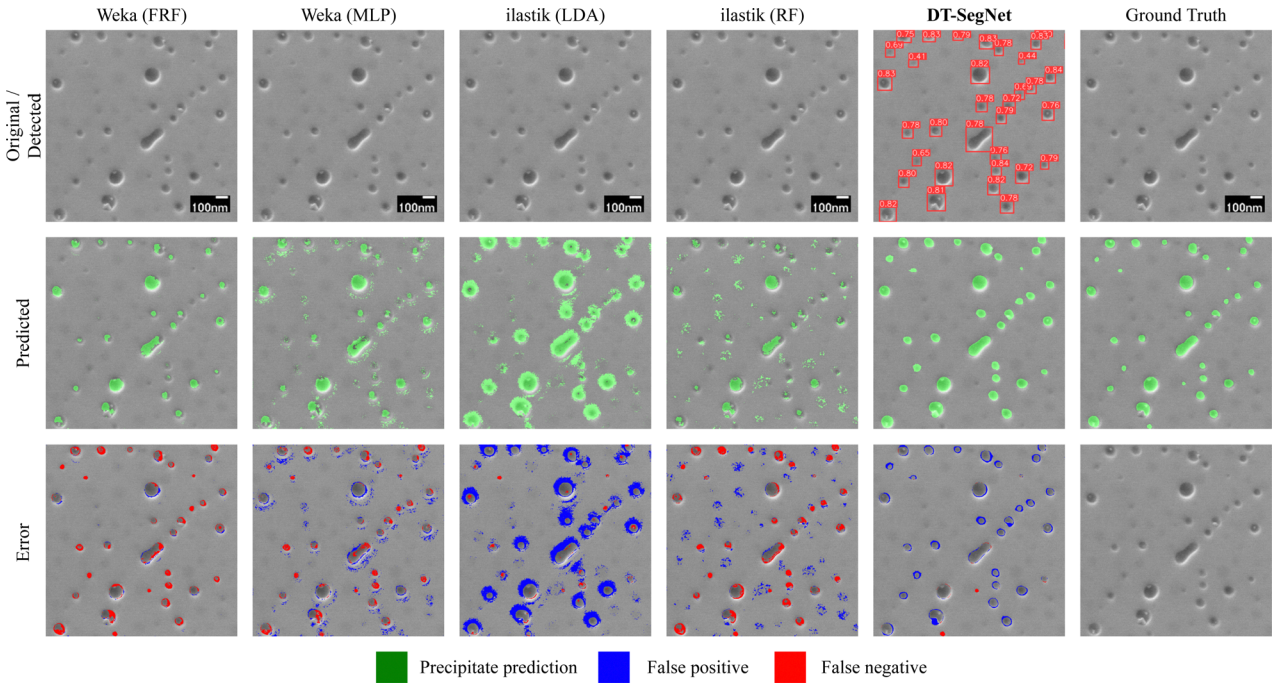


Fig. 8 Visualisation of segmentation results on 5-5-10 produced by four competing methods and this study's methods, along with the ground truth annotation.

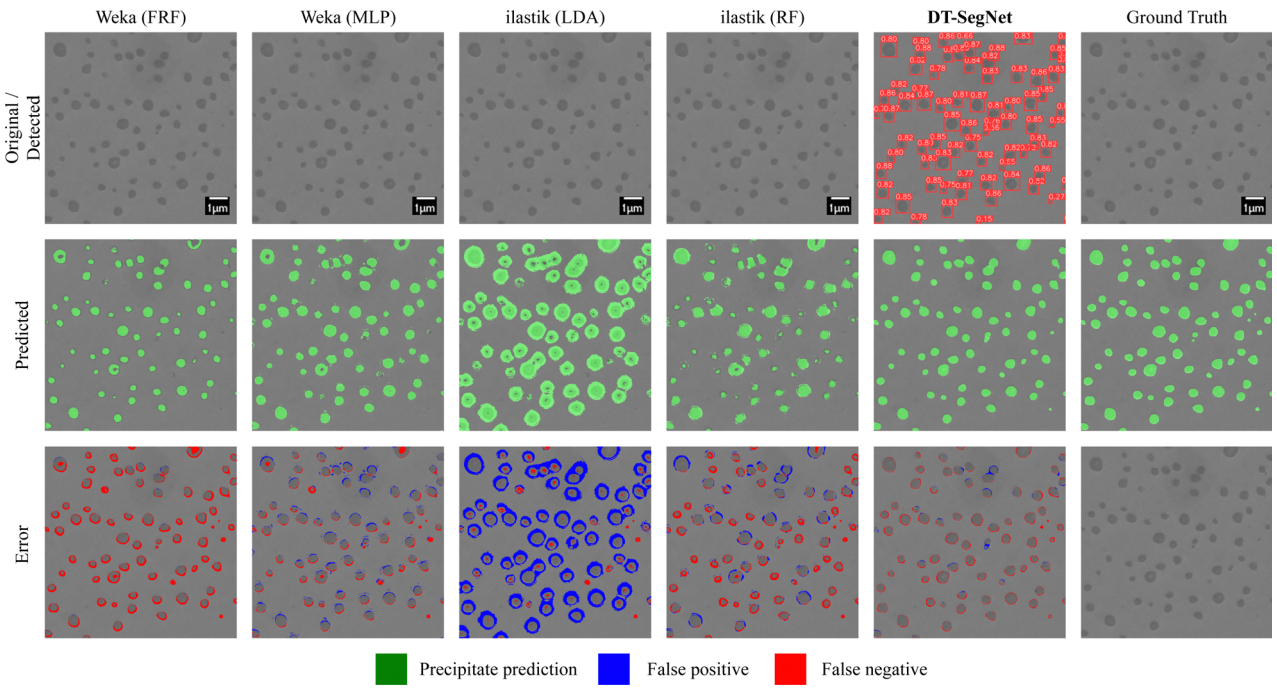


Fig. 9 Visualisation of segmentation results on 10-10-20-4 h produced by four competing methods and this study's methods, along with the ground truth annotation.

the exotic contrast in some precipitates due to the lack of robustness, which will affect the area measurement. The unstable interactive labelling mechanism of Weka and ilastik can cause this inability. On the other hand, DT-SegNet shows a substantially more accurate segmentation, achieving the lowest error rate of 2.28%.

Fig. 10 shows a case of SESEM images with micro-scale precipitates. Despite the evident edges of precipitates, the

 

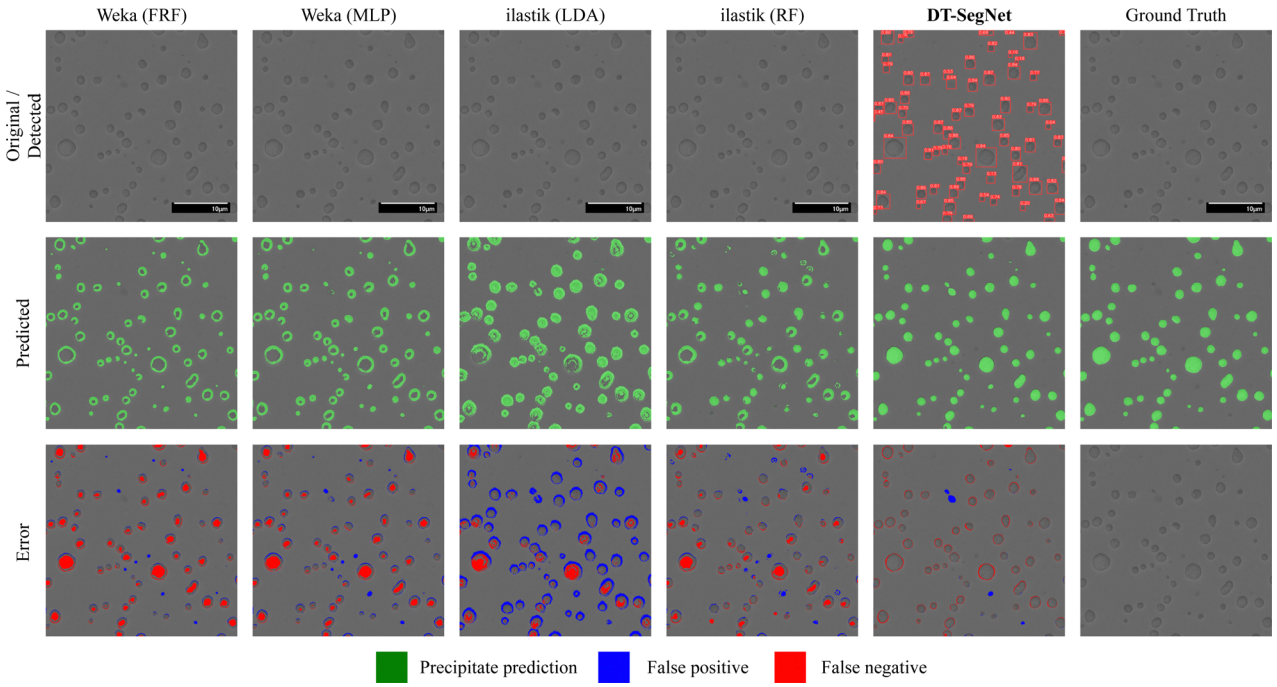


Fig. 10 Visualisation of segmentation results on 10-10-20-100 h produced by four competing methods and this study's methods, along with the ground truth annotation.

contrast inside these precipitates is similar to the matrix. In this case, all four baseline models manage to detect the edges but show poor segmentation results on the textures inside the precipitates, showing error rates higher than 3.5%. Since the segmentation network in DT-SegNet can capture most of the features, textures are well taken into account in the segmentation model, resulting in an outstanding performance of a 1.53% error rate.

The online computational time of DT-SegNet averaged on the test dataset is shown in Table 7. The manual segmentation time is estimated for EM images with 100 to 200 objects. It can be clearly seen that the proposed DT-SegNet can considerably improve the efficiency of precipitate segmentation compared to a manual process.

Overall, the proposed DT-SegNet considerably outperforms all Weka- and ilastik-based state-of-the-art approaches for multi-scale precipitate detection and area measurement from SEM images along with various background contrast.

### 4.9 Microstructural analysis of Cr-superalloys

Table 8 presents the results of the area fraction and average radius of precipitates measured manually (ground truth) and using the proposed DT-SegNet method. The two measurements are in good agreement, as discussed in the previous section. Here it is assumed that the volume fraction of precipitates equals the area fraction. It is worth noting that the two 10-10-20 alloys have higher precipitate volume fraction than the 5-5 and 5-5-10. The volume fraction is a key factor in pursuing high strength in these superalloys, as the precipitate strengthening, including ordering, coherency, modulus, and Orowan strengthening, increases with volume fraction.[71–75] Meanwhile, 10-10-20-4 h has smaller precipitates than 10-10-20-100 h due to the precipitate coarsening at 1200 °C.

Furthermore, analogous to some ferritic superalloys (Fe–NiAl systems) with a similar structure as the Cr–NiAl alloys,[76,77] it is assumed that the precipitates in these Cr-superalloys underwent diffusion-controlled coarsening during the used heat treatment condition. The particle size distribution (PSD) is plotted in Fig. 11. The co-ordinates are the probability density $\rho^2 h(\rho)$ which is calculated as:

$$\rho^2 h(\rho) = \frac{N_{r,r+\Delta r}}{\sum N_{r,r+\Delta r}} \frac{\bar{r}}{\Delta r} \quad (9)$$

Table 7 Process time of the proposed process and brute force manual. The manual segmentation time is estimated for EM images with 100 to 200 objects

| DT-SegNet | | | Manual |
|---|---|---|---|
| Detection | Segmentation | Total | Total |
| 0.0214 s | 1.8148 s | 2.3718 s | ≈30 min |

Table 8 Area fraction and average radius of precipitates by manual measurements and by DT-SegNet

| | Area fraction (%) | | Radius (nm) | |
|---|---|---|---|---|
| Image | DT-SegNet | Ground truth | DT-SegNet | Ground truth |
| 5-5 | 8.92 | 6.47 | 37.30 ± 9.85 | 32.66 ± 6.63 |
| 5-5-10 | 5.33 | 3.72 | 34.86 ± 11.54 | 29.33 ± 11.31 |
| 10-10-20-4 h | 8.99 | 10.00 | 210.09 ± 63.89 | 229.86 ± 63.60 |
| 10-10-20-100 h | 10.03 | 11.66 | 695.61 ± 267.50 | 752.23 ± 287.76 |

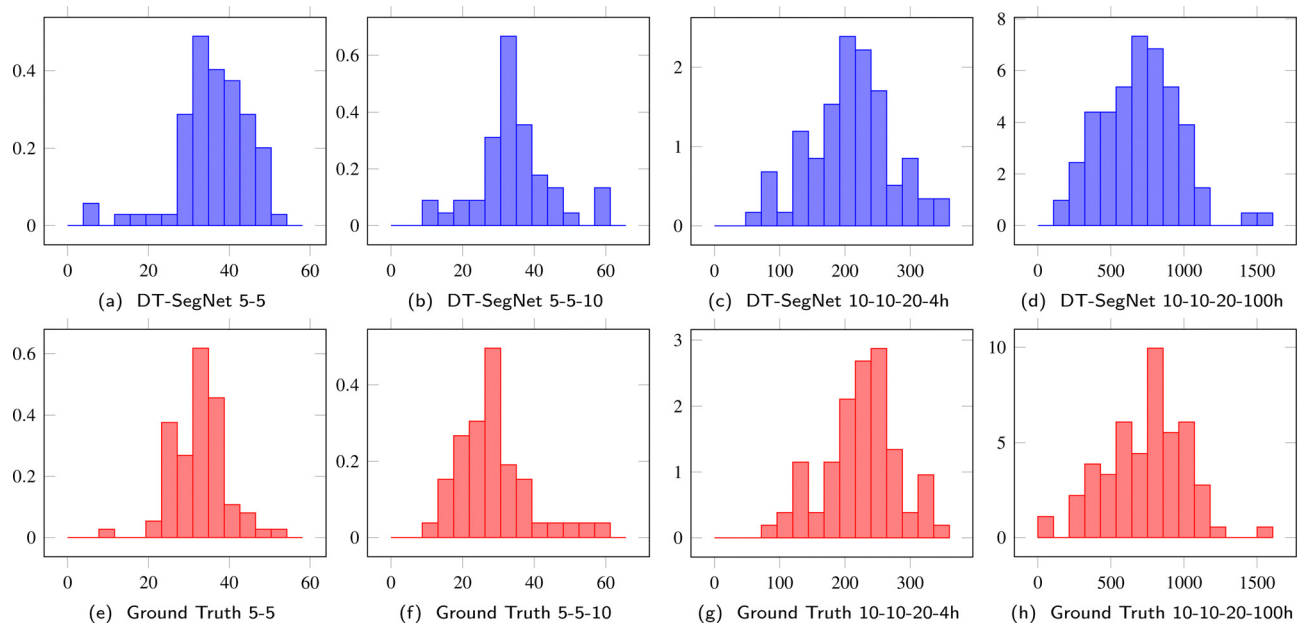


Fig. 11 The $\rho^2 h(\rho)$ particle size distribution of the four studied materials.

where $N_{r,r+\Delta r}$ is the number of precipitate in each interval, $\bar{r}$ is the average radius of precipitates and $\Delta r$ is the bin size of the distribution analysis. The two 10-10-20 alloys show a larger average radius suggesting a higher coarsening rate in 10-10-20 alloys than the 5-5 and 5-5-10. Along with the ageing, 10-10-20-100 h shows a broader distribution than 10-10-20-4 h as a result of precipitate coarsening, as observed in other A2–B2 systems like Fe–NiAl alloys.[36,37,76,77]

It is also worth noting that the ground truth only provides reference values for comparison among different segmentation methods and could be user-dependent. The measured values of the precipitate area and radius from SEM images by all methods are systematically smaller than their absolute values as the area of precipitates exposed to the surface is systematically smaller or equal to the largest cross-section of the precipitate sphere. Geometric correction for radius could be used to correct this bias.[37,39] Other frequently used imaging techniques, such as TEM could also provide similar measurements with different biases. The application of the current detection and segmentation method would also be of great interest for precipitate size analysis by TEM.

## 5 Conclusion

Efficient and accurate object detection, as well as segmentation, are important for EM image analysis when developing novel materials, and are critical to handle the large datasets associated with high-throughput combinatorial discovery methods. Traditional approaches consist of filtering EM images with a contrast threshold. However, the robustness of such a method can be challenged under different experimental conditions/noises, and often requires laborious manual adjustments.

In this work, a two-stage end-to-end deep learning scheme, DT-SegNet using state-of-the-art deep learning frameworks is proposed, namely YOLOv5 for object detection and Segformer for segmentation.

The model has been applied for precipitate pixel segmentation in novel Cr-superalloys, which comprise a two-phase microstruxture of an A2 Cr matrix with B2 NiAl spherical precipitates, developed for high-temperature applications such as advanced Concentrated Solar Power. The precipitates size and volume fraction are important factors controlling the mechanical properties in the superalloys. Extensive numerical experiments have shown the strength of DT-SegNet compared to the state-of-the-art tools Weka and ilastik in a number of different metrics, including accuracy, standard deviation, Recall, $F1$-score and SSIM. Furthermore, DT-SegNet is only trained using 15 images in this application. Thus, the proposed approach can be easily applied/transferred to other materials using a small amount of data for fine-tuning. The DT-Segment method is applied in the development of new Cr(Fe)–NiAl alloys for high-temperature applications. Area fraction, average radius and size distribution of precipitates were measured in different alloys where the precipitate size varies from nano-scale to micro-scale. In this multi-scale measurement, results from the DT-SegNet method show a good agreement with the manual measurement.

Future efforts can be considered to train the neural networks of detection and segmentation jointly so that the model fine-tuning for new materials can be further simplified. The tuned model will be further used for the determination of the precipitate coarsening rate of Cr-superalloys by measuring the precipitate size in function of the ageing time for a given temperature. The current training dataset can be expanded to datasets including not only Cr-superalloys but also other

 

advanced alloy systems, accelerating alloy development and microstructure examination. Furthermore, such low user intervention models are critical tools to enable the analysis of large datasets from high-throughput combinatorial metallurgy.

## Code and data availability

The computational part of this study is performed using Python language. The code and the EM data used in this study are available at: **https://doi.org/10.5281/zenodo.7510032**.

## Acronyms

| | |
|---|---|
| AP | Average precision |
| bcc | Body-centred-cubic |
| fcc | Face-centred-cubic |
| CALPHAD | CALculation of PHAse diagram |
| CNN | Convolutional neural network |
| COCO | Common objects in context |
| Cr | Chromium |
| CSP | Cross stage partial |
| DNN | Deep neural network |
| EM | Electron microscopy |
| FCNN | Fully convolutional neural network |
| FRF | Fast random forest |
| ICME | Integrated computational materials engineering |
| IoU | Intersection over union |
| LDA | Linear discriminant analysis |
| mAP | Mean average precision |
| MLP | Multi-layer perceptron |
| MGI | Materials genome initiative |
| Fe | Iron |
| NiAl | Nickel–aluminide |
| PANet | Path aggregation network |
| PRC | Precision-recall curve |
| RF | Random forest |
| ROI | Region of interest |
| SE | Secondary electron |
| SEM | Scanning electron microscope |
| SESEM | Secondary electron scanning electron microscope |
| SGD | Stochastic gradient descent |
| SPP | Spatial pyramid pooling |
| SPPF | Spatial pyramid pooling fast |
| SSIM | Structural similarity index |
| SVM | Support vector machine |
| SVC | Support vector machines C-support |
| TEM | Transmission electron microscopy |
| ViT | Vision transformer |
| YOLO | You only look once |

## Conflicts of interest

The authors have no conflicts of interest to disclose.

## Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 958418 "COMPASsCO2" (**https://www.compassco2.eu**). The authors thank the Centre for Electron Microscopy (University of Birmingham) for their support & assistance in this work. This work is partially supported by the EP/T000414/1 PREdictive Modelling with Quantification of UncERtainty for MultiphasE Systems (PREMIERE).

## References

1 S. Berg, D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, K. Eren, J. I. Cervantes, B. Xu, F. Beuttenmueller, A. Wolny, C. Zhang, U. Koethe, F. A. Hamprecht and A. Kreshuk, *Nat. Methods*, 2019, **16**, 1226–1232.
2 S. Curtarolo, G. L. W. Hart, M. B. Nardelli, N. Mingo, S. Sanvito and O. Levy, *Nat. Mater.*, 2013, **12**, 191–201.
3 W. Huang, P. Martin and H. L. Zhuang, *Acta Mater.*, 2019, **169**, 225–236.
4 M. Ge, F. Su, Z. Zhao and D. Su, *Mater. Today Nano*, 2020, **11**, 100087.
5 S. M. Hartig, *Curr. Protoc. Mol. Biol.*, 2013, **102**(1), 14–15.
6 W. B. Lievers and A. K. Pilkey, *Mater. Sci. Eng., A*, 2004, **381**, 134–142.
7 V. Amandine, M. Cédric, M. Sergio and D. Patricia, *Micron*, 2019, **121**, 90–98.
8 R. Sarma and Y. K. Gupta, *IOP Conf. Ser.: Mater. Sci. Eng.*, 2021, **1022**, 012027.
9 D. Ershov, M.-S. Phan, J. W. Pylvänäinen, S. U. Rigaud, L. Le Blanc, A. Charles-Orszag, J. R. W. Conway, R. F. Laine, N. H. Roy, D. Bonazzi, G. Duménil, G. Jacquemet and J.-Y. Tinevez, *Nat. Methods*, 2022, **19**, 829–832.
10 B. Nisha and M. Victor Jose, *Int. J. Adv. Res.*, 2018, **4**, 262–265.
11 X. Lu, W. Quan, S. Gao, G. Zhang, K. Feng, G. Lin and J. X. Chen, *IEEE trans. Intell. Transp. Syst.*, 2022, **23**, 15922–15939.
12 Q. Zhou, Z. Feng, Q. Gu, J. Pang, G. Cheng, X. Lu, J. Shi and L. Ma, *Context-Aware Mixup for Domain Adaptive Semantic Segmentation*, 2022.
13 W. Wang, X. Tan, P. Zhang and X. Wang, *IEEE J. Sel. Top. Quantum Electron.*, 2022, **15**, 6817–6825.
14 S. Cheng, I. C. Prentice, Y. Huang, Y. Jin, Y.-K. Guo and R. Arcucci, *J. Comput. Phys.*, 2022, 111302.
15 S. Cheng, Y. Jin, S. P. Harrison, C. Quilodrán-Casas, I. C. Prentice, Y.-K. Guo and R. Arcucci, *Remote Sens.*, 2022, **14**, 3228.
16 E. A. Holm, R. Cohn, N. Gao, A. R. Kitahara, T. P. Matson, B. Lei and S. R. Yarasi, *Metall. Mater. Trans. A*, 2020, **51**, 5985–5999.
17 B. L. DeCost and E. A. Holm, *Comput. Mater. Sci.*, 2015, **110**, 126–133.
18 S. M. Azimi, D. Britz, M. Engstler, M. Fritz and F. Mücklich, *Sci. Rep.*, 2018, **8**, 2128.
19 B. L. DeCost, B. Lei, T. Francis and E. A. Holm, *Microsc. Microanal.*, 2019, **25**, 21–29.

20 B. Ma, X. Ban, H. Huang, Y. Chen, W. Liu and Y. Zhi, *Symmetry*, 2018, **10**, 107.
21 G. Roberts, S. Y. Haile, R. Sainju, D. J. Edwards, B. Hutchinson and Y. Zhu, *Sci. Rep.*, 2019, **9**, 12744.
22 R. Cohn, I. Anderson, T. Prost, J. Tiarks, E. White and E. Holm, *JOM*, 2021, **73**, 2159–2172.
23 P. Liu, H. Huang, X. Jiang, Y. Zhang, T. Omori, T. Lookman and Y. Su, *Acta Mater.*, 2022, **235**, 118101.
24 Y. Wang, M. Lu, Z. Wang, J. Liu, L. Xu, Z. Qin, Z. Wang, B. Wang, F. Liu and J. Wang, *Mater. Des.*, 2021, **206**, 109747.
25 N. Wang, H. Guan, J. Wang, J. Zhou, W. Gao, W. Jiang, Y. Zhang and Z. Zhang, *Mater. Today Commun.*, 2022, **33**, 104954.
26 C. Sommer, C. Straehle, U. Köthe and F. A. Hamprecht, 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI), 2011, pp. 230–233.
27 I. Arganda-Carreras, V. Kaynig, C. Rueden, K. W. Eliceiri, J. Schindelin, A. Cardona and H. Sebastian Seung, *Bioinformatics*, 2017, **33**, 2424–2426.
28 R. C. Reed, *The Superalloys: Fundamentals and Applications*, Cambridge University Press, 2008.
29 W. D. Callister and D. G. Rethwisch, *Fundamentals of Materials Science and Engineering*, Wiley London, 2000.
30 A. M. Ges, O. Fornaro and H. A. Palacio, *Mater. Sci. Eng., A*, 2007, **458**, 96–100.
31 S. Zhao, X. Xie, G. D. Smith and S. J. Patel, *Mater. Lett.*, 2004, **58**, 1784–1787.
32 S. Meher, S. Nag, J. Tiley, A. Goel and R. Banerjee, *Acta Mater.*, 2013, **61**, 4266–4276.
33 D. J. Sauza, D. C. Dunand and D. N. Seidman, *Acta Mater.*, 2019, **174**, 427–438.
34 Ö. Dogan, X. Song, D. Palacio and M. Gao, *J. Mater. Sci.*, 2014, **49**, 805–810.
35 D. Locq, P. Caron, C. Ramusat and R. Mévrel, *Mater. Sci. Eng., A*, 2015, **647**, 322–332.
36 H. J. Dorantes-Rosales, V. M. Lopez-Hirata, J. L. Gonzalez-Velazquez, N. Cayetano-Castro and M. L. Saucedo-Muñoz, *Superalloys*, IntechOpen, 2015, p. 77.
37 Z. Sun, G. Song, J. Ilavsky, G. Ghosh and P. K. Liaw, *Sci. Rep.*, 2015, **5**, 16081.
38 Ö. Dogan, X. Song, S. Chen and M. Gao, *Intermetallics*, 2013, **35**, 33–40.
39 S.-I. Baik, M. J. S. Rawlings and D. C. Dunand, *Acta Mater.*, 2018, **153**, 126–135.
40 G. Song, Z. Sun, L. Li, X. Xu, M. Rawlings, C. H. Liebscher, B. Clausen, J. Poplawsky, D. N. Leonard, S. Huang, Z. Teng, C. T. Liu, M. D. Asta, Y. Gao, D. C. Dunand, G. Ghosh, M. Chen, M. E. Fine and P. K. Liaw, *Sci. Rep.*, 2015, **5**, 16327.
41 Z. Sun, C. H. Liebscher, S. Huang, Z. Teng, G. Song, G. Wang, M. Asta, M. Rawlings, M. E. Fine and P. K. Liaw, *Scr. Mater.*, 2013, **68**, 384–388.
42 G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, Y. Kwon, T. Xie, K. Michael, J. Fang, imyhxy, Lorna, C. Wong, Z. Yifu, A. V. , D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, tkianai, yxNONG, P. Skalski, A. Hogan, M. Strobel, M. Jain, L. Mammana and Xylieong, NanoCode012, Ultralytics/YOLOv5: V6.2 – YOLOv5 Classification Models, Apple M1, Reproducibility, Clearml and Deci.Ai Integrations, 2022, **https://zenodo.org/record/7002879**.
43 E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez and P. Luo, *Advances in Neural Information Processing Systems (NeurIPS)*, 2021, pp. 12077–12090.
44 K. He, G. Gkioxari, P. Dollár and R. Girshick, *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, 2017, pp. 2961–2969.
45 A. Krizhevsky, I. Sutskever and G. E. Hinton, *Commun. ACM*, 2017, **60**, 84–90.
46 Z.-Q. Zhao, P. Zheng, S.-T. Xu and X. Wu, *IEEE Trans. Neural Netw. Learn. Syst.*, 2019, **30**, 3212–3232.
47 J. Redmon, S. Divvala, R. Girshick and A. Farhadi, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
48 J. Redmon and A. Farhadi, YOLOv3: An Incremental Improvement, *aXiv*, 2018, preprint, arXiv:1804.02767, DOI: **10.48550/arXiv.1804.02767**.
49 A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, *arXiv*, 2020, preprint, arXiv:2004.10934, DOI: **10.48550/arXiv.2004.10934**.
50 C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh and I.-H. Yeh, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, 2020, pp. 390–391.
51 S. Liu, L. Qi, H. Qin, J. Shi and J. Jia, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8759–8768.
52 S. Ioffe and C. Szegedy, *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.
53 H. Zhang, M. Cisse, Y. N. Dauphin and D. Lopez-Paz, *Mixup: Beyond Empirical Risk Minimization*, 2018.
54 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
55 A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale, *arXiv*, 2020, preprint, arXiv:2010.11929, DOI: **10.48550/arXiv.2010.11929**.
56 Y. Liu, L. Chu, G. Chen, Z. Wu, Z. Chen, B. Lai and Y. Hao, PaddleSeg: A High-Efficient Development Toolkit for Image Segmentation, *arXiv*, 2021, preprint, arXiv:2101.06175, DOI: **10.48550/arXiv.2101.06175**.
57 D. Tzutalin, *Labelimg*, 2015, **https://github.com/tzutalin/labelImg**.
58 T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, *European Conference on Computer Vision (ECCV)*, Cham, 2014, pp. 740–755.
59 O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, *ImageNet Large Scale Visual Recognition Challenge*, 2015.

60 R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Pearson, New York, NY, 2018.
61 L. Breiman, *Mach. Learn.*, 2001, **45**, 5–32.
62 M. Kubat, *Knowl. Eng. Rev.*, 1999, **13**, 409–412.
63 T. Hastie, R. Tibshirani and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2009, vol. 2.
64 J. Platt, *Adv. Large Margin Classifiers*, 1999, **10**, 61–74.
65 O. Ronneberger, P. Fischer and T. Brox, *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Cham, 2015, pp. 234–241.
66 H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen and J. Wu, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 1055–1059.
67 L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff and H. Adam, *European Conference on Computer Vision (ECCV)*, Cham, 2018, pp. 833–851.
68 M. Everingham and J. Winn, *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep.*, 2011, vol. 8, p. 5.
69 Z. Wang and A. C. Bovik, *IEEE Signal Process. Mag.*, 2009, **26**, 98–117.
70 W. Luo, Y. Li, R. Urtasun and R. Zemel, *Advances in Neural Information Processing Systems (NIPS)*, 2016.
71 R. B. Schwarz and R. Labusch, *J. Appl. Phys.*, 1978, **49**, 5174–5187.
72 B. Reppich, *Acta Mater.*, 1998, **46**, 61–67.
73 L. M. Brown and W. M. Stobbs, *Philos. Mag. (1798–1977)*, 1971, **23**, 1201–1233.
74 E. Nembach, *Scr. Metall.*, 1984, **18**, 105–110.
75 U. F. Kocks, *Mater. Sci. Eng.*, 1977, **27**, 291–298.
76 N. Cayetano-Castro, M. L. Saucedo-Muñoz, H. J. Dorantes-Rosales, J. L. Gonzalez-Velazquez, J. D. Villegas-Cardenas and V. M. Lopez-Hirata, *Adv. Mater. Sci. Eng.*, 2015, **2015**, e485626.
77 H. Calderon and M. E. Fine, *Mater. Sci. Eng.*, 1984, **63**, 197–208.