# 4

# An End-to-End AI-based Automated Process for Semiconductor Device Parameter Extraction

**Dinu Purice[1], Matthias Ludwig[2], and Claus Lenz[1]**

[1]Cognition Factory GmbH, Germany
[2]Infineon Technologies AG, Germany

## Abstract

In this work, we present an automated AI-supported end-to-end technology validation pipeline aiming to increase trust in semiconductor devices by enabling a check of their authenticity. The high revenue associated with the semiconductor industry makes it vulnerable to counterfeiting activities potentially endangering safety, reliability and trust of critical systems such as highly automated cars, cloud, Internet of Things, connectivity, space, defence and supercomputers [7]. The proposed approach combines semiconductor device-intrinsic features extracted by artificial neural networks with domain expert knowledge in a pipeline of two stages: (i) a semantic segmentation stage based on a modular cascaded U-Net architecture to extract spatial and geometric information, and (ii) a parameter extraction stage to identify the technology fingerprint using a clustering approach. An in-depth evaluation and comparison of several artificial neural network architectures has been performed to find the most suitable solution for this task. The final results validate the taken approach, with deviations close to acceptable levels as defined by existing standards within the industry.

**Keywords:** Semantic segmentation, image processing, hardware trust, physical inspection of electronics, AI, ML, deep learning, supervised learning, convolutional neural networks, computer vision.

## 4.1 Introduction

Automation is one of the key parameters industries can approach to strengthen quality and lower overall costs. The improved availability of data and the mainstream application of approaches relying on artificial intelligence (AI) pushes industries towards the adaption of these AI methods. Nonetheless, practical implementations of these often seem to fail due to inflated expectations. Via a use-case from the semiconductor industry, we show various practical ways to overcome these potential pitfalls.

The recently introduced European Chips act recognises the paramount importance of the semiconductor industry within the global economy. The market for integrated electronics was at $452.25B in 2021 and is expected to grow to $803.15B in 2028 [8]. The high revenue potential causes extreme cost pressure and a highly competitive market. Consequently, since decades, the semiconductor industry is driven to automation along the complete value chain. One way to differentiate from competitors is through the utilisation of AI-powered manufacturing enhancements which have the potential to gain $35B - $40B annually over the entire industry [10]. Yet, not only manufacturing yields the potential to benefit from the industries push towards AI. The methods also offer the chance to be used for trust generation. In the aforementioned staggering market, rogues also aim to catch their share through counterfeiting, i.e. cloning, remarking, overproducing, or simply reselling of used parts [9]. This leads to the use case discussed throughout this work: via physical inspection and a fully integrated AI flow we present a fully automated assessment of the technological properties of a device. The idea for such a pipeline has already been introduced in [15] where it is argued that through a subsequent analysis of the cross-sections, the authenticity of the manufacturing technology can be validated. Relevant features in this case include geometric shapes and dimensions of the constituent structures, as well as material-related properties. Each technology can be interpreted as an individual fingerprint, such that deviations from specifications can be reported as suspicious. This work will focus on the end-to-end application aspects of the use case and includes following contributions:

- We will introduce an end-to-end, fully automated flow for semiconductor device technological parameter extraction by image segmentation and pattern recognition as an exemplary industrial use-case.
- We introduce our methodology that is tailored to the requirements of the use case. This includes an image segmentation approach which is

constituted of a set of specialised U-net cascades, class-specific loss functions, and an evolution-based training approach.
- The advantages of our design-decisions are quantitatively compared to similar state-of-the-art approaches and important lessons learned – transferable to other use-cases – are summarised.

**Related work:** The demand for measuring structures and critical dimensions within semiconductor devices is ever-increasing. While manufacturing relies mostly on in-line metrology, a further possibility is the post-production measurement. The databases are oftentimes big and automating of these flows is vital. A first template-based approach has been shown in [30]. This work relies on template matching and pattern recognition for the extraction of profile parameters. Furthermore, in a previous work [15], we have proposed how the flow can be utilised for the detection of counterfeit electronics [9] by comparing the extracted parameters against a database of known parameters.

The prospect of (semi)-automation of industrial processes through the use of machine learning-based (ML) methods is further gaining traction due to recent advancements in the field of ML and the uncovering of its unprecedented feature extraction and generalisation capabilities. Further accelerated due to the abundance of data, the "smartisation" of industrial processes through ML techniques has been conceived as the fourth industrial revolution [6].

The data set involved in this application bears two important characteristics: it consists of grey-valued images, and more importantly has a very limited availability of annotated data. The same characteristics are typically observed in medical applications, in dealing with images produced by computed tomography (CT), cone beam computed tomography (CBCT), as well as magnetic resonance imaging (MRI), ultrasound, X-ray, all data types being scarcely available to the public due to the confidential nature of medical data. Nevertheless segmentation tasks have been successfully tackled by ML-based methods, and in particular deep learning approaches which were proven to satisfy the high accuracy requirements typical to applications in the medical field. Of particular note in this context is the work of Ronneberger *et al.* [22] with the introduction of the U-net, a symmetric network consisting of a encoding and a decoding arm which was proved to possess high generalisation capabilities even on relatively small data sets. The progress was further accentuated after the debut of Dice-based loss functions, first introduced by Milletari *et al.* [17], which have been proven to outperform existing alternatives in the analysis of highly skewed data. Based

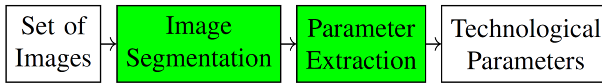


**Figure 4.1** Overview of the architecture.

on the above-mentioned innovations, both supervised and unsupervised deep learning-based approaches have been constantly expanding within different use cases in the medical field, as shown by the works of Kawula *et al.* [11], Wang *et al.* [3] or Altaf *et al.* [2].

The following chapters describe the two paramount steps of this application, namely the Image Segmentation and the Parameter Extraction stages, respectively. Both stages are currently being fine-tuned and validated to ensure compliance with industry-defined standards of operation.

## 4.2 Semantic Segmentation

### 4.2.1 Proof of Concept and Architecture Overview

As a first step of development a benchmark stage was conducted, with the goal of determining the viability of an AI-based approach to scanning electron microscope (SEM) image segmentation and identify the most suitable architecture for the task. Considering that both the industrial sector and the academic sector lack openly available annotated semiconductor cross-section SEM data, a custom data set was assembled and labelled. The data set consists of 1024 by 685 grey-valued images, obtained at Infineon Technologies AG's failure analysis laboratories and represent technology nodes from 500 nm to approximately 40 nm with copper and Al-Tu technologies included. Devices with less than one metal layer (e.g. discrete transistors) were excluded. The image sources are state-of-the-art SEMs available in semiconductor failure analysis laboratories. For the purpose of this stage 202 images were manually sampled and labelled.

The images were annotated with 5 relevant labels of interest, namely "metal", "VIA", "lateral isolation", "poly", and "deep trench isolation" [25], each bearing features important in the process of technology identification. The selected features imbue the following purposes within a semiconductor device:

- **Metal:** Low resistance metallic connections between devices. Several metallisation layer can be stacked over each other to route inter-device connections.

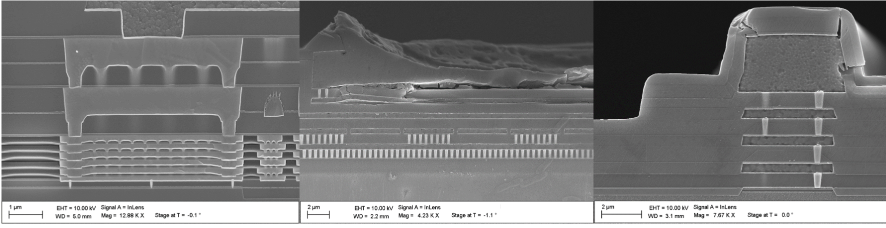

**Figure 4.2** Examples showcasing different semiconductor technologies

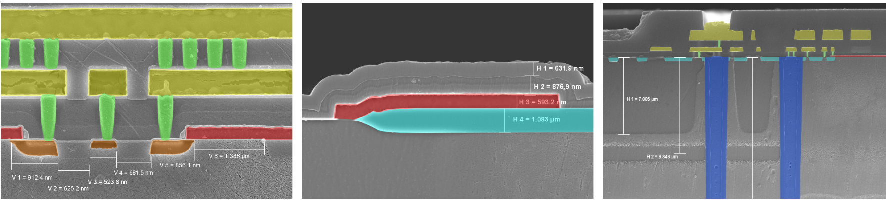

**Figure 4.3** Examples of labelled data showcasing the different ROIs: green – VIA; yellow – metal; teal – lateral isolation; red – poly; blue – deep trench isolation

- **Vertical interconnect access (VIA) / contact:** Low ohmic interconnections between different metallisation layers (VIA) or between devices and the lowest metallisation layer.
- **Lateral isolation (shallow trench isolation):** Electrical lateral isolation between devices with a dioxide trough a *shallow* deposition.
- **Deep trench isolation:** Trenches for lateral isolation with a high depth-width ratio. Mostly found in analogue integrated circuits.
- **Poly:** Poly-crystalline silicon which is used as gate electrode.

For the benchmark stage however only two regions of interest (ROIs) were selected, namely "VIA" and "metal". The two ROIs strongly differ in terms of size and quantity, with the pixel-wise class-distribution of the "metal" objects representing 13.61%, while "VIA" objects being more numerous but at the same time smaller, taking up 2.5%. Therefore, they reflect the two important properties of the expected data: high variability and high skewness.

As it can be seen in Fig. 4.4 there is a strong overlap in intensity between the various regions of interest, yielding classical segmentation methods such as thresholding [21], region-growing [20], watershed [18] and k-means clustering [19] ineffective. Instead, an effective segmentation process requires

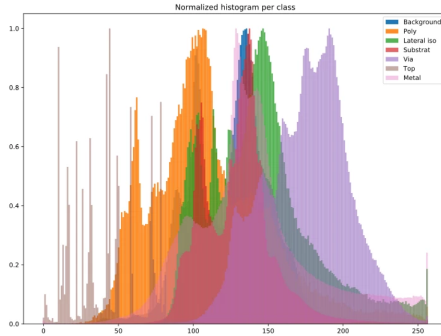


**Figure 4.4** Histograms of the investigated data grouped by label of interest

domain-expert knowledge – thus encouraging the use of deep learning-based methods capable of extracting spatial and semantic features. Several network architectures were selected as candidates, based on their respective performance in similar segmentation tasks. An overview of each candidate network architecture is presented below:

- **U-net** [22]

Introduced by Ronneberger *et al.* [22] as a solution for biomedical image segmentation, this architecture has been shown to perform reasonably well even when trained with small amounts of data. It consists of an down-sampling encoder and an up-sampling decoder arm enabling efficient spatial context capture. The arms are connected with skip connection which accelerate convergence during training and combat vanishing gradients. The U-net achieved an averaged Dice score of 0.76 on the test subset.

- **Feature Pyramid Network (FPN)** [13]

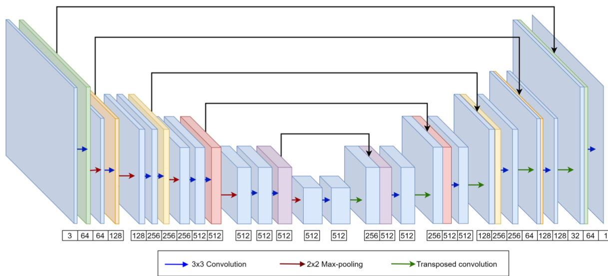


**Figure 4.5** Overview of the U-net architecture [24]

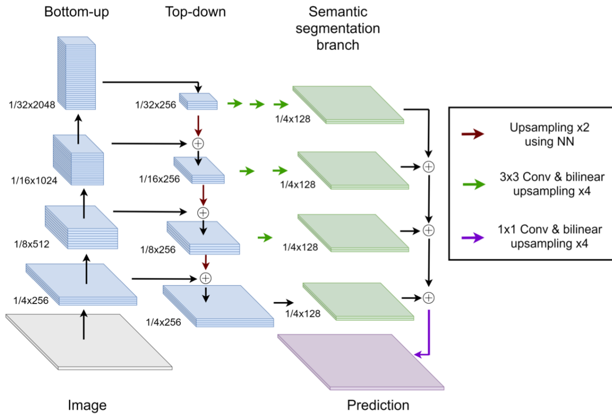


**Figure 4.6** Overview of the FPN architecture [24]

The FPN follows a top-down approach with skip connections, similar to the previously mentioned U-net. However instead of using the final output as the prediction, the FPN makes predictions for each stage (see Fig. 4.6) thus combining semantically strong low-resolution features with semantically weaker high-level features. An additional segmentation branch is used to then merge the information from all levels into a single output. The FPN obtained an averaged Dice score of 0.71 on the test subset

- **Gated-Shape Convolutional Neural Network (GSCNN)** [27]

The GSCNN employs a two-stream architecture, with the shape-related features focused in a dedicated stream that works in parallel to the standard

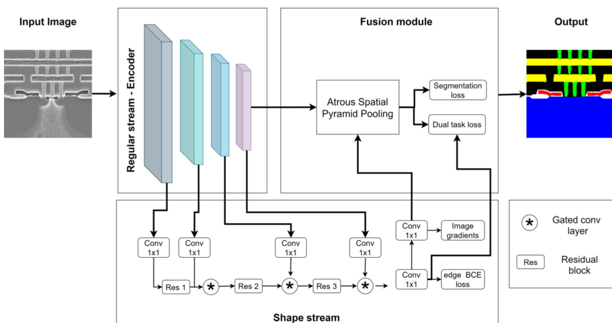


**Figure 4.7** Overview of the GSCNN architecture [24]

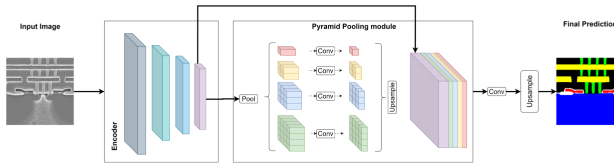


**Figure 4.8** Overview of the PSPNet architecture [24]

encoder. A key characteristic of this architecture is the use of gated convolutional layers, which connect intermediate layers of both streams, facilitating the transfer of information from the encoder to the shape stream while filtering irrelevant information. The information of both streams is then combined within the fusion stage using an Atrous Spatial Pyramid Pooling module (ASPP). An averaged Dice score of 0.74 on the test subset was obtained by the GSCNN.

- **Pyramid Scene Parsing Network (PSPNet)** [31]

The PSPNet architecture makes use of a Pyramid Pooling Module (PPM) to capture rich context information from the output of the encoder arm. The capture is done through fusion of the network's four pyramid scales, as seen in Fig. 4.8. An averaged Dice score of 0.69 on the test subset was obtained using the PSPNet architecture.

- **Siamese network** [16]

The Siamese network presents another approach to combine the features extracted at low-resolution and high-resolution levels, namely through a two step approach. The first step operates on the whole, down-sampled image, and outputs a coarse segmentation map. As a second step the segmentation

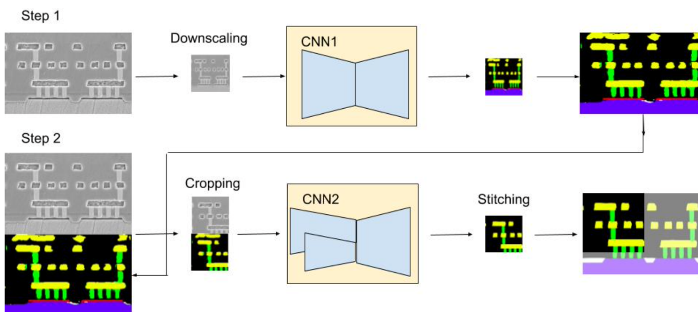


**Figure 4.9** Overview of the Siamese network architecture [12]

map is then fed into a Siamese network containing two encoders (as show in Fig. 4.9), with the original high resolution image going through the other encoder in patches. Finally the decoder stitches together the patches, obtaining a segmentation map at the same resolution as the input image. The Siamese network reached an averaged Dice score of 0.78 on the test subset.

### 4.2.2 Implementation Details and Result Overview

To complete the benchmark stage, each network architecture was trained 5 times on random pre-sampled splits of the data set (60% training, 20% validation, 20% test). The resulting Dice scores (averaged over the 5 tries and the 2 labels of interest) and their respective spread is presented in Fig. 4.10 below. All experiments were ran on a server equipped with Intel Core i9-9940x (14 Cores, 3,30GHz), 4 RTX 5000 GPUs and 128 GB RAM.

The two best performing approaches are the Siamese (with a mean Dice score of 0.78) and the U-net (with a mean Dice score of 0.76). The performance of the Siamese approach can be explained by the two steps

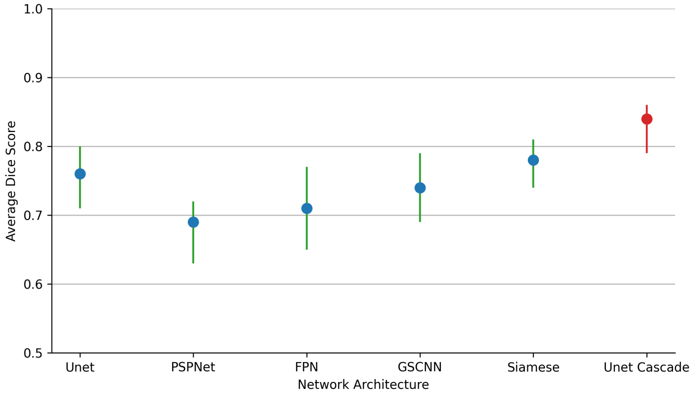


**Figure 4.10** Average Dice Scores (blue) and spread (green) per investigated network architecture, along with the final chosen architecture (red)

**Table 4.1** Obtained Dice Scores for each showcased network architecture

| Architecture | U-net | PSPNet | FPN | GSCNN | Siamese |
|---|---|---|---|---|---|
| Average DSC | 0.76 | 0.69 | 0.71 | 0.74 | 0.78 |
| DSC range | 0.71 - 0.80 | 0.63 - 0.72 | 0.65 - 0.77 | 0.69 - 0.79 | 0.74 - 0.81 |

analysis employed by this method, which segments firstly in low resolution therefore with a larger perceptive field, followed by a second step analysing in a higher resolution, with the downside of having a lower perceptive field. On the other hand, the U-net architecture obtained similar results with much lower resource consumption during training and inference.

Based on this performance, a branched U-net cascade was chosen as the preferred architecture, combining both the two step analysis at different resolution levels, as well as the generalisation power associated with the U-net. The chosen architecture consists of independent branches targeting each ROI. For each branch, a 2D U-net takes the down-sampled image as input and produces an intermediate, rough segmentation, which is then up-sampled to the dimensions of the original input image. The intermediate segmentation is then aggregated with the original high-resolution input image to be fed into a 3D U-net (as introduced by Milletari *et al.* [17]), which then outputs a high-resolution segmentation map. Some practical advantages of such a modular architecture are the possibility to update each branch individually in case of additional data being available, as well as to allow scaling up with additional branches targeting new labels without having to update each branch. An overview of the described architecture for a given branch is presented in Fig. 4.11. Repeating the experiment in benchmark conditions has yielded an averaged Dice score of 0.84 (shown in red in Fig. 4.10), outperforming all the other candidate architectures.

Typical to deep learning applications with limited data sets and despite the use of data augmentation techniques, overfitting was proven to be an issue. This could be clearly seen in the discrepancy between the Dice scores on the train and test subsets respectively. Having chosen an architecture for the fine-tuning stage, additional effort was invested in expanding the data set from 202 to 2192 images. For this stage of the application all five previously mentioned labels of interest were trained on.

Due to the relatively large number of hyper-parameters to be tuned a population-based training method was used, consisting of two evolution phases: exploration and exploitation. During the exploration phase the networks are trained with randomly sampled hyper-parameters sets. During the following exploitation phase the best performing sets of hyper-parameters are identified, and new sets are sampled in close proximity within the hyper-parameter space.

Although Dice loss has proven itself effective in segmentation tasks, the high skewness and variability as well as low availability of data require additional compensatory mechanisms. For this purpose several alternative

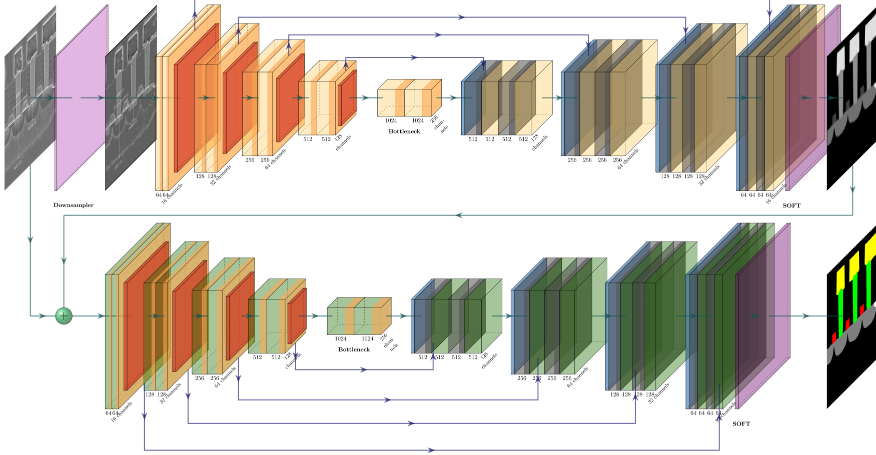


**Figure 4.11** An overview of the U-net cascade architecture, consisting of a 2D U-net (top) and a 3D U-net (bottom) which takes as input the high resolution input image stacked with the output segmentation of the first stage

loss functions were investigated as hyper-parameters, including Focal Tversky loss [1], Combo Loss [26], Unified Focal Loss (LogCoshDSC) [29]. Training experiments indicated that the loss function is the paramount hyper-parameter, having the most impact upon the resulting accuracy of the network. Furthermore, different labels have been shown to benefit differently from each loss function. For example the network trained on the "metal" label, which has the highest pixel-wise distribution of all classes and typically large structures on each image, performed best when trained using the LogCoshDSC loss. At the same time the "VIA" and "poly" labels, both with a very low pixel-wise distribution ($< 2.5\%$) were segmented best by networks trained with the Focal Tversky loss. The Combo loss on the other hand was most effective for the networks targeting the "lateral iso" and "deep trench" labels, which have an average pixel-wise distribution but are the most difficult to identify visually.

The average Dice scores obtained on the test set for each label of interest are presented in the table below.

The "metal" and "VIA" labels obtained the highest Dice scores, with a substantial increase in accuracy of about 10% compared to the benchmark stage. Also of particular note is the "deep trench" case. Despite being the class with the lowest pixel-wise distribution, only appearing in 58 of the images,

**Table 4.2** Averaged Dice Scores for each label of interest

| Label | Metal | VIA | Poly | Lateral iso | Deep trench |
|---|---|---|---|---|---|
| Loss function | LogCosh | Foc. Tversky | Foc. Tversky | Combo | Combo |
| Average DSC | 0.93 | 0.91 | 0.88 | 0.82 | 0.76 |

the proposed network architecture was able to segment it with reasonable accuracy to make use of the extracted information.

## 4.3 Parameter Extraction

The process following the semantic image segmentation is the extraction of the technological device parameters. The overview of the algorithmic approach is shown in algorithm 1. The inputs are the image meta-data – with the sole relevant information being the pixel size per image – and the segmented image. In a first step the segmented are written to polygon while retaining their class-labels. Subsequently, the polygons of every class (C) are retrieved. From this set of polygons, polygons below a statistically evaluated threshold (area of a polygon instance lower than five times the mean of a polygon instances within this class) are removed from the list From these *cleaned* polygons, the centroids of the single objects are computed which are utilised for clustering. The customised clustering method is shown in table 4.3.

**Algorithm 1** Parameter extraction approach

▷ Input: Segmented image and corresponding meta-data

1: **function** ParameterExtraction($Image, Meta$)
2: $Polygons_C \leftarrow GetPolygons(Image, Meta)$
3: **for all** $Classes$ **do**
4: $Polygons_C \leftarrow GetPolygonsOfSingleClass(Image)$
5: $Polygons_C \leftarrow CleanPolygons(Polygons_C)$
6: $Polygons_{C,i} \leftarrow ClusterPolygonsVertically(Polygons_C.Centroid)$
7: **for all** $Clusters$ **do**
8: $Polygons_{C,i,j} \leftarrow ClusterPolygonsHorizontally(Polygons_{C,i})$
9: $Parameters_{C,i} \leftarrow ExtractParameters(Polygons_{C,i,j})$
10: **end for**
11: **end for**
12: **return** $Parameters$
13: **end function**

**Table 4.3** Utilised cluster evaluation techniques [14]. Notation: $n$: number of objects in data-set; $c$: centre of data-set; $NC$: number of clusters; $C_i$: the i-th cluster; $n_i$: number of objects in $C_i$; $c_i$: centre of $C_i$; $W_k$: the within-cluster sum of squared distances from cluster mean; $W*_k$ appropriate null reference; $B$ reference data-sets

| Method | Definition | Value |
|---|---|---|
| CH [4] | $\frac{\sum_i n_i \cdot d^2(c_i, c)/(NC-1)}{\sum_i \sum_{x\in C_i} d^2(x, c_i)/(n-NC)}$ | Elbow |
| Gap [28] | $log\left(\frac{(\prod W^*_{kb})^{1/B}}{W_k}\right)$ | Elbow |
| DB [5] | $\frac{1}{NC} \cdot \sum_i \left\{ \frac{\frac{1}{n_i}\sum_{x\in C_i} d(x, c_i) + \frac{1}{n_j}\sum_{x\in C_j} d(x, c_j)}{d(c_i, c_j)} \right\}$ | Min |
| DB$^2$ [5] | $\frac{1}{NC} \cdot \sum_i \left\{ \frac{\frac{1}{n_i}\sum_{x\in C_i} d(x, c_i) + \frac{1}{n_j}\sum_{x\in C_j} d(x, c_j)}{d(c_i, c_j)} \cdot i^2 \right\}$ | Min |
| Sil. [23] | $\frac{1}{NC} \sum_i \left\{ \frac{1}{n_i} \sum_{x\in C_i} \frac{b(x)-a(x)}{max[b(x), a(x)]} \right\}$ | Max |

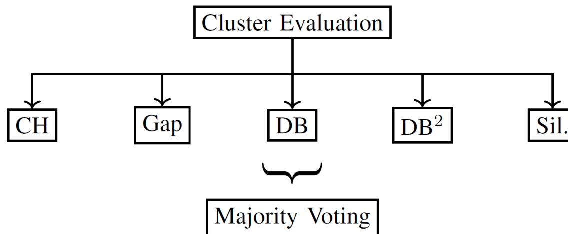


**Figure 4.12** Utilised cluster evaluation techniques.

Different cluster evaluation techniques – namely Calinski-Harabasz (CH) [4], gap [28], Davies-Bouldin (DB) [5], a custom squared Davies-Bouldin (DB$^2$) [5], and silhouette (Sil.) [23] – are conducted on one-dimensional feature vectors which are constituted of the y-share of the centroid coordinates. The previous clustering is done via trough k-means clustering while the k is kept – adapted to the use case – between *2* and *10*. For the different evaluation techniques optimal number of clusters ($k$) are reported through different metrics (minimum, maximum, elbow). This computationally costly approach is suitable for the use case since the vectors are one-dimensional and the total number of polygon objects to be evaluated is relatively small ($< 100$). In the final step of the vertical clustering, the optimal number of

clusters is inferred through a majority voting among the individual evaluation techniques.

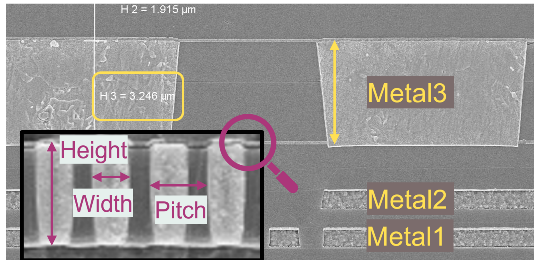


**Figure 4.13** Example cross-section image with annotated metal and contact/VIA features

Since the polygon objects are now vertically assigned, a clustering in the horizontal dimension is the next step. The procedure is the same as previously discussed for the vertical clustering. For the vertically and horizontally clustered elements, the technological, geometrical parameters can be inferred. These are illustrated via figure 4.13 for the metal and VIA classes. The vertical height is determined for metallisation layers and height, width, and pitches for the interconnecting contact and VIA layers. After the polygons objects are assigned to classes, these attributes can be calculated through trivial mathematical operation. Height is the difference of the bounding box maximum and minimum in vertical dimension. Width is the difference of the bounding box in horizontal dimension, and pitches are the differences of the x-coordinate of the centroid of two adjacent polygon objects. The values are respectively averaged within all classes.

An example is shown for the VIA class through the example in figure 4.14. After segmentation of the grey-scale image, the individual segmented classes are inferred into polygon objects. Here the VIA class is exemplified. The vertical clustering process is shown through the two right images. The dendrogram visualises the linkage of the different clusters which are subsequently optimised via discussed approach. The optimum number of clusters are shown in the bottom right figure. An evaluation techniques report an optimum of four (different values constitute the optimum) clusters. Following this, these four are subsequently clustered in horizontal dimension and respectively geometrically inferred. Following results were obtained for this example (besides the absolute values, the relative deviation to a manual measurement is given):

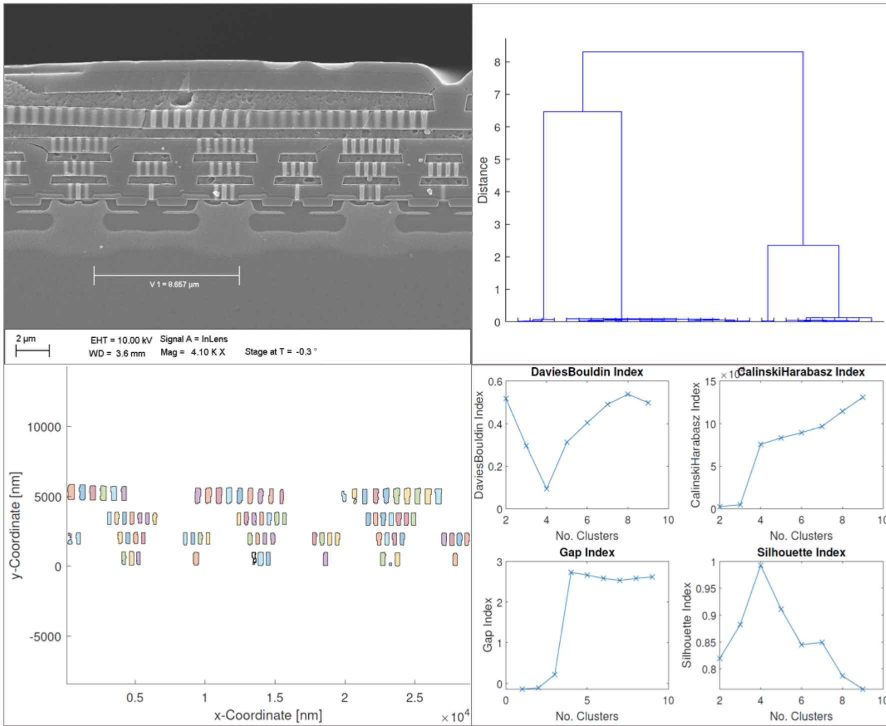


**Figure 4.14** Example cross-section image (upper left). The polygonised VIA objects are shown (lower left). A dendrogram is shown for the relative distances of the y-coordinates of the single objects (upper right). Finally, the results of the utilised cluster evaluation techniques are presented (lower right).

- Contact (**h**eight, **w**idth, **p**itch): 942 nm (+9.66%), 319 nm (+5.98%), 525 nm (+12.42%)
- VIA1 (h, w, p): 870 nm (+26.82%), 319 nm (n.a.), 545 nm (n.a.)
- VIA2 (h, w, p): 898 nm (+14.83%), 319 nm (n.a.), 542 nm (n.a.)
- VIA3 (h, w, p): 1086 nm (+16.27%), 434 nm (n.a.), 750 nm (n.a.)

Within the technology validation use case, the inferred technological features are tested against the designed and manufactured technological properties. This is computed via multi-dimensional distance matching (e.g. Euclidean, rectilinear distance) of both vectors. The validation accuracy depends on several different factors which are the segmentation quality, parameter extraction accuracy and image acquisition completeness. Experiments have shown

that the current automated end-to-end flow reaches 75% accuracy for previously known Al-Tu technologies. Improvement is necessary for copper (Cu) technologies which are more complex to segment. According to existing procedures within the industry, deviations of less than 5% for pitches and deviations of less than 25% for all other geometrical measurements compared to a ground truth, i.e. the designed technology parameters are acceptable. The same requirements have been used as a benchmark for the validation of this application. The high deviations are a consequence attributed to process variances during device manufacturing and de-processing. Presented image shows a single frame which was acquired in a sub-optimal zoom level for measuring discussed features. Yet, almost all requirements were achieved. In summary it can be stated that the proof-of-concept presented in this work displays strong potential to satisfy existing industrial requirements, especially when adequate zooms levels are chosen for the particular technological parameters.

## 4.4 Conclusion

The settings for AI implementation in an industrial setting are often completely different from consumer applications. Data being scarce the design of productive AI application is forcibly *data-driven*, or more specifically *data-adapted*. Industrial parameters are manifold, and the requirements typically impose the need to automate, improve, or even enable new processes. To make an AI-based solution *viable* these requirements must be met. In this work, we have shown through an end-to-end technology demonstrator – incorporating deep learning and cluster evaluation – showcasing the automation of semiconductor technology identification based on SEM cross-section analysis. A comparison of different convolutional neural network architectures was presented, and a candidate best suited for the SEM segmentation task was drafted. The proposed candidate architecture represents a cascade of 2D and 3D Unets, arranged in branches each dedicated to a single label of interest. Following a pragmatic perspective, a modular design is proposed, ensuring scalability and ease-of-maintenance. Trained on a custom-created data set of 2192 images, the proposed architecture obtained Dice scores in the range of 0.76-0.93 for labels of different complexity, arguing in favour of the employment of supervised deep learning-based methods even in applications with strongly limited amounts of available labelled data. Based on the obtained results, a parameter extraction algorithm is proposed, aimed at exploiting the obtained segmentation maps with the purpose of identifying

and validating the technology of the investigated semiconductor devices. The obtained results were in the range of ground truth measurements with deviations in an acceptable measuring range. The potential for narrowing down these uncertainty ranges were outlined.

## 4.5 Future Work

Following the development and validation steps described above, a production test stage will determine the potential of the segmentation component of the process to be used in other applications of semiconductor analysis. Aside from a high degree of automation and the mandatory fulfilment of functional requirements, industry has established high thresholds for non-functional requirements. Maintainability, system up time, extensibility, usability, or updateability are just some of the potential requirements across different industries. Such requirements are addressed by specialised frameworks such as Ray and TorchServe. Combined with the advantages of a modular architecture, they enable the possibility to update each network with virtually no down-time. Additional investigations are conducted in the expansion of the data augmentation pipeline, with the goal of increasing the exploitation of the available data set regardless of its relatively small size.

### Acknowledgment

This work is conducted under the framework of the ECSEL AI4DI "Artificial Intelligence for Digitising Industry" project. The project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826060. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Austria, Czech Republic, Italy, Latvia, Belgium, Lithuania, France, Greece, Finland, Norway.

## References

[1] N. Abraham and N. Mefraz Khan. A novel focal tversky loss function with improved attention u-net for lesion segmentation. *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 683–687, 2019.

[2] F. Altaf, S. M. S. Islam, N. Akhtar, and N. Khalid Janjua. Going deep in medical image analysis: Concepts, methods, challenges, and future directions. *IEEE Access*, 7:99540–99572, 2019.

[3] S. Budd, E. C. Robinson, and B. Kainz. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis*, 71:102062, 2021.

[4] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3(1):1–27, 01 1974.

[5] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[6] A. Diez-Olivan, J. Del Ser, D. Galar, and B. Sierra. Data fusion and machine learning for industrial prognosis: Trends and perspectives towards industry 4.0. *Information Fusion*, 50:92–111, 2019.

[7] European Commision. A chips act for europe.

[8] Fortune Business Insights. Semiconductor market size, share & covid-19 impact analysis, 2021-2028. FBI102365.

[9] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris. Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain. *Proceedings of the IEEE*, 102(8):1207–1228, 2014.

[10] S. Göke, K. Staight, and R. Vrijen. Scaling ai in the sector that enables it: Lessons for semiconductor-device makers.

[11] M. Kawula, D. Purice, M. Li, G. Vivar, S.-A. Ahmadi, K. Parodi, C. Belka, G. Landry, and C. Kurz. Dosimetric impact of deep learning-based ct auto-segmentation on radiation therapy treatment planning for prostate cancer. *Radiation Oncology*, 17, 01 2022.

[12] G.-M. Konnerth. Exploring application-oriented methods to improve cnn-based segmentation of sem microchip images. Master's thesis, Technical University of Munich, 2020.

[13] X. Li, T. Lai, S. Wang, Q. Chen, C. Yang, R. Chen, J. Lin, and F. Zheng. Weighted feature pyramid networks for object detection. In *2019 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 1500–1504, 2019.

[14] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. Understanding of internal clustering validation measures. In *2010 IEEE International Conference on Data Mining*. IEEE.

[15] M. Ludwig, B. Lippmann, A.-C. Bette, and C. Lenz. Demo: A fully automated process for semiconductor technology

analysis through SEM cross-sections. In *25th International Conference on Pattern Recognition (ICPR)*.

[16] K. Martin, N. Windunga, S. Sani, S. Massie, and J. Clos. A convolutional siamese network for developing similarity knowledge in the selfback dataset, 2017.

[17] F. Milletari, N. Navab, and S.-A. Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 10 2016.

[18] L. Najman and M. Schmitt. Watershed of a continuous function. *Signal Processing*, 38(1):99–112, 1994. Mathematical Morphology and its Applications to Signal Processing.

[19] D. Nameirakpam, K. Singh, and Y. Chanu. Image segmentation using k -means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54:764–771, 12 2015.

[20] R. Nock and F. Nielsen. Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1452–1458, 2004.

[21] N. Otsu. A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9:62–66, 1979.

[22] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[23] P. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, November 1987.

[24] F. Schiegg. Boundary detection and semantic segmentation of sem-images. Master's thesis, Technical University of Munich, 2020.

[25] S. M. Sze and K. K. Ng. *Physics of Semiconductor Devices*. Wiley, 2006.

[26] S. A. Taghanaki, Y. Zheng, S. K. Zhou, B. Georgescu, P. Sharma, D. Xu, D. Comaniciu, and G. Hamarneh. Combo loss: Handling input and output imbalance in multi-organ segmentation. *Computerized Medical Imaging and Graphics*, 75:24–33, 2019.

[27] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5228–5237, 2019.

[28] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. 63(2):411–423.
[29] M. Yeung, E. Sala, C.-B. Schönlieb, and L. Rundo. A mixed focal loss function for handling class imbalanced medical image segmentation. *ArXiv*, abs/2102.04525, 2021.
[30] X. Zhang, Z. Fu, Y. Huang, A. Lin, Y. Shi, and Y. Xu. Effective method to automatically measure the profile parameters of integrated circuit from SEM/TEM/STEM images. In *2017 China Semiconductor Technology International Conference (CSTIC)*. IEEE, 3 2017.
[31] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, 2017.