Image Retrieval in Semiconductor Manufacturing

No Author Given

No Institute Given

Abstract. Content-Based Image Retrieval is gaining importance in many industrial scenarios, where large collections of data from manufacturing need to be automatically queried e.g. for quality inspection purposes. In this work we design an image retrieval solution over IMAGO, a dataset of Transmission Electron Microscopy (TEM) images of nano-sized silicon structures collected in the production site of STMicroelectronics, in Agrate Brianza, Italy. This dataset presents significant idiosyncrasies including: *i)* only a limited portion of images are provided with labels, where class information refer to specific parts of electronic components exhibiting similar structures, *ii*) annotations cover only a few classes, and many images refer to unseen classes that are not represented in the training set, and *iii)* images of the same class can be very dissimilar as these are acquired at different magnification levels of the electronic microscope. Our main contribution is the design of an effective image retrieval system based on a deep neural network. In particular, we present a novel training procedure that alternates between siamese loss, assessed on annotated samples, and reconstruction loss, assessed on unlabelled samples. In this way we exploit the whole information in the IMAGO dataset, obtaining a single network which is able to effectively retrieve images of unseen classes without overfitting the known ones. Our solution successfully outperforms all state-of-the-art approaches over the IMAGO dataset and it is currently being deployed in STMicroelectronics production sites.

Keywords: Content-Based Image Retrieval · Metric Learning · Siamese Networks · Autoencoders.

1 Introduction

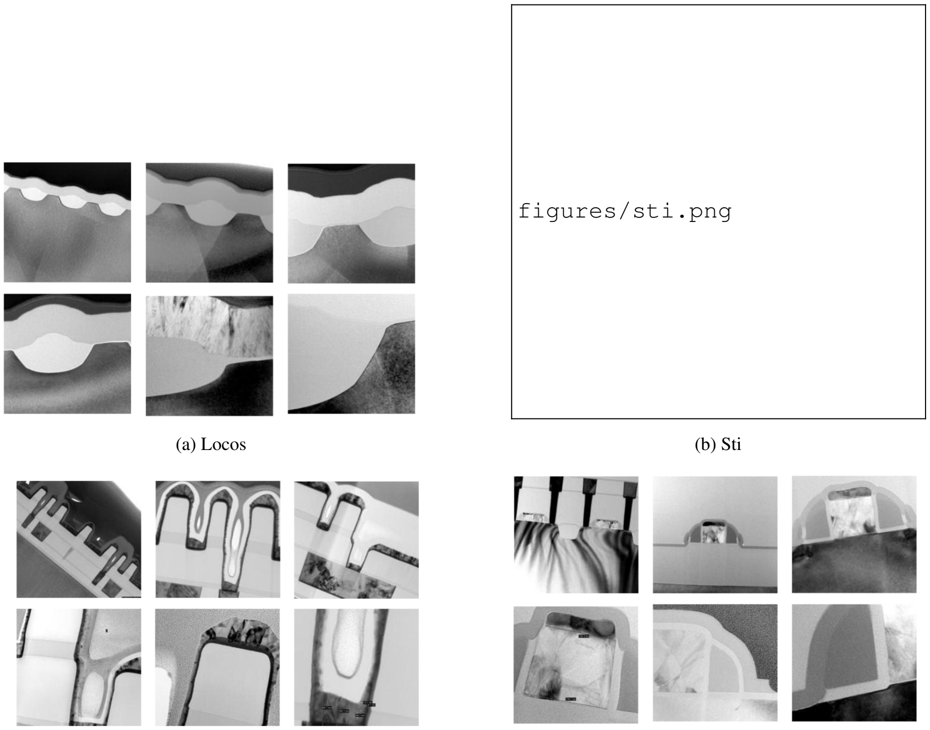
Pushed by a steadily increasing demand of chips and memories, semiconductor industries have been struggling to produce smaller electronic components to improve performance, power efficiency and device reliability, and the same time, reduce the production costs, times and wastes. Transmission Electron Microscopy (TEM) have played a very important role to improve semiconductor manufacturing processes, as it can yield highly magnified images (up to 2 million times). TEM images are used to perform the physical characterization and the compositional analysis of specimens and to analyze faults and nano-sized structures of semiconductor, both during the design and the production stages. In the production site of STMicroelectronics in Agrate Brianza, hundreds of TEM images are acquired everyday and collected in a large dataset called IMAGO, containing around two millions of entries acquired during different production stages and modalities, thus exhibiting variable quality and diverse magnifications and resolutions. Only a fraction of IMAGO images are associated to a known class, namely a *semiconductor structure*, respectively *locos*, *sti*, *spacers*, *tccv* and *cell* (Figure 1). Many images in IMAGO instead refer to *unknown* classes that have never been annotated, like those in Figure 2.

A tool enabling easy searches through the IMAGO dataset would be extremely useful to retrieve TEM images similar to a query one, and this would help engineers to set up new technologies, diagnose and source manufacturing problems, as well as train inexperienced personnel on the wide variety of structures observed. In particular, it is of paramount importance to retrieve images belonging to the same class of the query, which in our case means images referring to the same semiconductor structure. Retrieval should be performed whether or not the structure has been previously annotated.

Developing an image retrieval system over the IMAGO dataset raises a series of unique challenges that can make most image retrieval solutions from the literature not a viable option. In fact, only a very limited portion of IMAGO images are provided with labels indicating the class of a shown structure and, moreover, images of the same class can be very dissimilar when acquired at different magnification levels of the electronic microscope, as illustrated in Figure 1. Therefore, all the methods assuming that images belonging to the same class differ by transformations preserving a few distinctive visual features (e.g. change of perspective, noise, as in [2]), do not apply here. Moreover, training a siamese neural network to learn image similarity/dissimilarity from labels [3], [7], [13] might not be an option given the few annotated samples. Conversely, when the majority of images are not labelled, a valid strategy is to adopt an autoencoder to extract an embedding of the images [18], [15], [20], [11], as this can be trained in an unsupervised manner. Retrieval models based on autoencoders however might fail at identifying images that belong to same class but are visually dissimilar, as these might have embeddings that are very far apart, resulting in a poor variability in the retrieved images.

Here we present a system able to retrieve IMAGO images depicting the same structure, thus belonging to the same class, of a given query, disregarding whether this belongs to a class annotated in the training set or not. Our solution is based on a deep neural network trained by alternating the minimization of two losses: a (supervised) triplet loss typical of siamese neural networks and an (unsupervised) reconstruction loss typical of autoencoders. In this way, we exploit the whole information in the IMAGO dataset obtaining a single neural network trained, potentially, over the entire dataset. Then, image retrieval consists in simply extracting network embedding and performing similarity search in the dataset. Our system, thanks to the proposed training strategy, is able to effectively retrieve images even of unknown classes, i.e., representing structures that have never been annotated, without overfitting the small training set of labelled images. Additionally, we prove that our solution leads to outstanding retrieval results, outperforming three state-of-the-art alternatives. In fact, our experiments conducted over the IMAGO dataset confirm we achieve the best results both in terms of precision and mean average precision. Through a leave-one-class-out experiment, we show that we achieve stable results when querying unknown structures, and that our method provides a good variability in the scale of the retrieved images, exploiting not only the visual similarity between images but also the semantic information provided by the labels. The proposed solution is currently being deployed in the production site of STMicroelectronics in Agrate Brianza, Italy.

The rest of the paper is organized as follows. In Section 2 we describe the IMAGO dataset, and in Section 3 we review the literature of Content Based Image Retrieval. In Section 4 we formulate our specific industrial problem in mathematical terms. In Section 5 we present our solution, and in Section 6 we validate our claims through three detailed experiments. Finally, in Section 7 we draw the conclusions of our work and we mention some possible future works.



(a) Locos

(b) Sti

(c) Tccv

(d) Spacers

Fig. 1: Examples of labelled images in IMAGO. Images belonging to labelled classes representing four known structures (respectively locos, sti, tccv and spacers) are reported, each acquired at a different magnification (spanning from $0.1\mu m$ to $10nm$). Note the large variability in content, resolution, illumination and noise level of images belonging to the same class.

2 IMAGO dataset

The IMAGO dataset contains around two millions of TEM images that are acquired from a wide variety of processes and imaging modalities (variable quality, diverse magnifications and resolutions), as they were acquired by scientists and engineers for different purposes and on different microscopes. IMAGO presents some challenges that

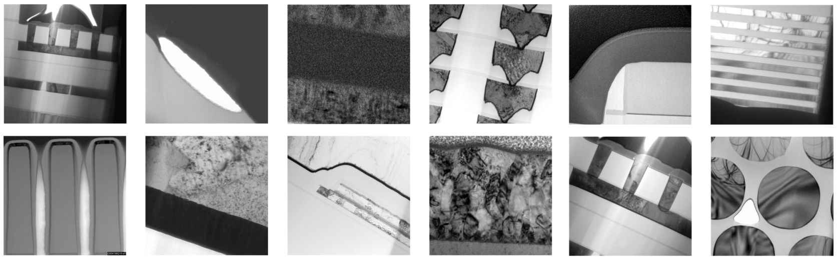


Fig. 2: Examples of images belonging to unknown classes, which represent the vast majority of the IMAGO dataset. Images shown here reports structures that have never been annotated.

prevent an effective retrieval by straightforwardly applying state-of-the-art solutions. In particular, only a small part of the dataset has been labelled (about 2500 images), providing information about the class which images belong to. This requires a specific semi-supervised approach in order to exploit all the available information during training. Moreover, the annotations provided cover only a few classes, while many images, like those illustrated in Figure 2, belong to unknown classes that need nevertheless to be taken into account during training. Finally, images are acquired under significantly different conditions, resulting in samples subject to transformations like rotations and shifts, and, more importantly, the different magnification at which images are acquired produces a large variety of scales even between samples belonging to the same class, as reported in Figure 1.

Among the many semiconductor structures represented in IMAGO, only five have been labelled, respectively locos, sti, spacers, tccv and cell. *LOCal Oxidation of Silicon* (locos) and *Shallow Trench Isolation* (sti) are isolation structures that separate MOS transistors preventing current leakage between adjacent components. Usually, locos is the isolation structure of choice for technology nodes larger than $250nm$, while sti is employed elsewhere. Spacers structures insulates the gate of each transistor from the adjacent drain and source contacts, while tccv structures arise in the damascene process used to create the copper interconnections between active elements of the wafer. Finally, cell indicates phase-change memory (PCM) cells, which are currently under development at STMicroelectronics. The wide majority of images in IMAGO dataset instead refer to other classes that are never been annotated. Thus, unlabelled images might either refer to unknown classes, as well as to one of the five previously described. It is important to stress that lack labels is not due to the inability of the experts to classify them, but simply depends on the dataset size which makes the manual annotation of each image unfeasible. Therefore, an expert or a pool of experts is always able to associate each image to its corresponding class by visual inspection.

3 Related Works

Image retrieval methods can be divided in two main approaches: *concept-based* and *content-based*. In the first case, images are manually annotated by text descriptors, which are then used by a dataset management system to perform retrieval [8]. Concept-based techniques date back to 1970s and have become outdated given the progressive increase in the amount of collected data, preventing textual annotation of each image. *Content-Based Image Retrieval* is certainly more appealing as it enables searching through a large database of images without any textual keywords but simply using a query image.The mainstream approach to perform content-based image retriaval consists in extracting feature vectors from the query image and then assessing its similarity with the feature vectors extracted from the dataset images to retrieve the closest images.

Content-based image retrieval suffers from the so called *semantic gap*, namely the "lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation" [19]. Depending on how wide the semantic gap is, three different CBIR tasks can be identified [5]. *Duplicate retrieval* searches for variants of the query image which might have undergone geometric or photometric transformations, like image cropping, scaling and adjustments in colors, brightness, or contrast. *Instance retrieval*, searches for images that depict the same instance of an object under different conditions, like illumination, viewing angle, scale (e.g., the Eiffel tower under different weather conditions). *Semantic retrieval* covers a spectrum broader than instance retrieval and searches for images belonging to the same class as the query (e.g., dog, cat). Observe that our specific industrial problem falls within the scope of the CBIR task of *semantic retrieval*, which aims for finding images belonging to the same category as the query. In our specific industrial context two TEM images belongs to the same category when the observed semiconductor structure is the same, whether the structure has been annotated or not.

In the last decade CBIR has undergone a paradigm shift driven by the advent of Deep Learning. In the early days several solutions based on hand-crafted features like color, textures, shapes, and gradients were mainly employed [9]. After 2014, deep learning approaches which learn features directly from data have emerged as a dominating alternative [6], and a critical aspect is the loss function and used to train these models, which boils down to defining a similarity measure in the feature space. This is the context of *metric learning* methods [10] and both *supervised* and *unsupervised* methods have been proposed to learn embedding in the (latent) feature such that similar input samples are close together in the embedded space, while dissimilar ones are far apart according to a certain metric distance measure. Supervised approaches have access to a set of labelled inputs, while on the contrary, unsupervised approaches only take as input unlabelled data and aim to learn embedding enabling the reconstruction of the input. Nonetheless, given the peculiar nature of IMAGO, consisting of partially annotated data, we cannot employ a completely supervised or unsupervised approach, but need to find a balance between the two settings, exploiting the strengths of both. In this sense, our problem can be formulated as a *semi-supervised* image retrieval problem.

Here we review the most similar approaches to our solution. The first one belongs to the supervised image retrieval framework and employs a siamese network trained with triplet loss [3]. This approach has been widely used in different image retrieval scenarios where training data are provided with labels. Examples of this approach are [3], where a siamese network is trained to construct an embedding of medical images, [7], which is applied to public image datasets as MNIST, and [13], where the siamese network was developed and applied to a public dataset containing photographs (Flickr15k). However, as mentioned in Section 1, training a siamese network might become unfeasible when a complete knowledge on the similarity/dissimilarity of training data is missing, and when data are unlabelled. The second approach is unsupervised, and extracts the embedding from an autoencoder, trained to reconstruct input data from a lower dimensional representation [18]. While the autoencoder is particularly suited for an unsupervised setting, its limitations are the inability to exploit any knowledge about similarity or class membership during training. Methods based on this approach work mainly on unlabelled data, as in [15] where a dataset consisting of pictures belonging to ten unknown classes is considered for the retrieval, and in [20] and [11] where public datasets like Cifar-10 and ImageNet are employed without labels. All datasets considered in the solutions from the literature do not present any of the peculiarities of IMAGO, nor any solution presented is constructed specifically to deal with the semi-supervised settings we consider here and the large variation in terms of scale resulting from the acquisition process.

4 Problem Formulation

The problem we address in this paper is that of retrieving TEM images belonging to the same class of a given query image $\mathbf{q}$ from the IMAGO dataset. Let $\mathcal{D}$ denote the IMAGO dataset, which contains images at different resolutions. Each image $\mathbf{x} \in \mathcal{D}$ is associated with a class, corresponding to a specific structure $y \in \mathcal{Y} = \{y_1, \dots, y_C\}$, where C denotes the total number of structures in $\mathcal{D}$. Therefore, given a query image $\mathbf{q} \in \mathcal{D}$, our goal is to select K images $\{\mathbf{x}_1, \dots, \mathbf{x}_K\}$ from $\mathcal{D}$ such that each $\mathbf{x}_i$ belongs to the same class as $\mathbf{q}$.

We assume that only a small subset $\mathcal{T}$ of the dataset $\mathcal{D}$ containing annotated samples is available for training. Moreover, the classes of images in $\mathcal{T}$ belong to a subset of $\mathcal{Y}$, which we refer to as the known classes $\mathcal{S} \subset \mathcal{Y}$. However, our goal is to perform an effective retrieval either when the query image $\mathbf{q}$ belongs to a known class in $\mathbf{S}$ or to an unknown class $\mathcal{U} = \mathcal{Y} \backslash \mathcal{S}$.

5 Proposed Solution

We solve the retrieval of TEM images over the IMAGO dataset by learning an embedding function $f \colon \mathcal{D} \to \mathbb{R}^M$ that defines a dimensionality reduction procedure by mapping each image $\mathbf{x} \in \mathcal{D}$ into a feature vector $f(\mathbf{x})$ in the Euclidean space $\mathbb{R}^M$. We learn the embedding f through a neural network so that images from the same class correspond to feature vectors that are close to each other, while images from different classes are mapped in feature vectors that are far apart. Then, given a query image $\mathbf{q}$, we retrieve the images whose feature vectors are the closest to $f(\mathbf{q})$. The details of our training procedure are presented in Section 5.1, while in Section 5.2 we explain how we use the learned f to perform image retrieval. Finally, we report the implementation details of our solution in Section 5.3.

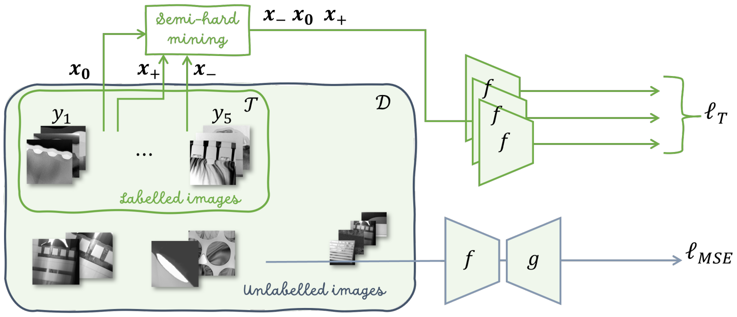


Fig. 3: Training phase of our solution. Semi-hard mining is employed to sample triplets from $\mathcal{T}$, the labelled part of $\mathcal{D}$, to train the siamese network by minimizing the triplet loss ℓ_T in (1). The siamese network is built by replicating f in three branches. This training procedure is alternated with the training of the autoencoder, obtained by stacking f and g, where all samples in $\mathcal{D}$ are employed and the reconstruction loss ℓ_{MSE} in (3) is minimized.

5.1 Learning the Embedding Function

To train our neural network f we design a specific training procedure where we employ both labelled and unlabelled images in $\mathcal{D}$, including samples unknown classes in $\mathcal{U}$. The proposed procedure is illustrated in Figure 3. At first, we build a siamese network by replicating f into three branches. This siamese network is fed with triplets of samples drawn from $\mathcal{T}$ using semi-hard mining [14] and is trained by minimizing a triplet loss ℓ_T in (1).

The main drawback of employing a siamese network is that, being trained on supervised samples, it would behave unpredictably on samples from unknown classes in $\mathcal{U}$, which are not represented in $\mathcal{T}$. For this reason we adopt a reconstruction loss ℓ_{MSE} in (3) to train our network f on all the samples in $\mathcal{D}$. To this purpose, we build an autoencoder, where f acts as an encoder and we adopt another neural network g as a decoder. During training we alternate the optimization of the two losses. More precisely, we alternate the following steps for a maximum number of iterations:

- train the siamese network by minimizing the triplet loss ℓ_T over $\mathcal{T}$ for an epoch;
- train the autoencoder (thus both f and g) by minimizing the reconstruction loss ℓ_{MSE} over $\mathcal{D}$ for an epoch.

After training the decoder g is discarded. In what follows we detail how these two steps are performed.

Training the Siamese Network. The minimization of the triplet loss [12], [20] is based on the extraction of triplet of samples from the training set $\mathcal{T}$. Given a randomly sampled anchor $\mathbf{x}_0 \in \mathcal{T}$, we extract a positive and a negative sample, $\mathbf{x}_+$ and $\mathbf{x}_-$, respectively such that $\mathbf{x}_0$ and $\mathbf{x}_+$ belong to the same class, while $\mathbf{x}_-$ belongs to a different one. The triplet loss over $(\mathbf{x}_0, \mathbf{x}_+, \mathbf{x}_-)$ is defined as

$$\ell_T(\mathbf{x}_0, \mathbf{x}_+, \mathbf{x}_-) = \max(\|f(\mathbf{x}_0) - f(\mathbf{x}_+)\|_2^2 - \|f(\mathbf{x}_0) - f(\mathbf{x}_-)\|_2^2 + R, 0)\,, \qquad (1)$$

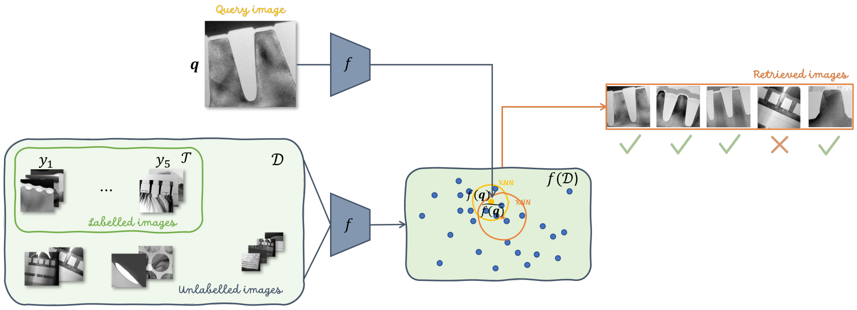


Fig. 4: Retrieval phase of our solution. Once the embedding function has been learned, we compute and store the feature vectors for all the images in the dataset $\mathcal{D}$. Given a query image $\mathbf{q}$, we first compute its embedding in $\mathbb{R}^M$ using the trained f, and then we perform query expansion. The first K samples closest to the expanded query are selected as the output of the system.

where the parameter $R > 0$ plays the role of a margin which measures how "far" we want samples that do not belong to the same class to be in our feature space. Optimizing f to minimize the triplet loss, we ensure that $f(\mathbf{x}_0)$ and $f(\mathbf{x}_+)$ are near-by, while $f(\mathbf{x}_0)$ and $f(\mathbf{x}_-)$ are pushed apart.

Different strategies can be employed to sample triplets $(\mathbf{x}_0, \mathbf{x}_+, \mathbf{x}_-)$ from the training set and in particular we adopt semi-hard mining [14]. In semi-hard mining, once the anchor $\mathbf{x}_0$ and the positive sample $\mathbf{x}_+$ have been drawn, the negative sample $\mathbf{x}_-$ is selected so that the distance in the feature space between $\mathbf{x}_0$ and $\mathbf{x}_-$ is greater than the distance between $\mathbf{x}_0$ and $\mathbf{x}_+$, but not large enough to nullify the loss function (1). In practice, $\mathbf{x}_-$ is selected such that:

$$\|f(\mathbf{x}_0) - f(\mathbf{x}_+)\|_2^2 < \|f(\mathbf{x}_0) - f(\mathbf{x}_-)\|_2^2 < \|f(\mathbf{x}_0) - f(\mathbf{x}_+)\|_2^2 + R. \qquad (2)$$

If no of such $\mathbf{x}_-$ is available in $\mathcal{T}$, we select $\mathbf{x}_-$ as the negative sample closest to $\mathbf{x}_0$, as explained in [14]. The other popular alternative to construct triples, hard-mining [16], selects negatives $\mathbf{x}_-$ that are instead closer to $\mathbf{x}_0$ rather than $\mathbf{x}_+$.

Training the Autoencoder. Autoencoders are widely employed neural networks for feature learning without supervision. Typically, autoencoders consist of two subnetworks, an *encoder* and a *decoder*. The encoder learns to represent each input sample in a latent space, while the decoder learns to reconstruct the input from its latent representation. Here, we adopt f as the encoder, thus the latent representation of an input image $\mathbf{x}$ is actually the feature vector $f(\mathbf{x})$. We train a second neural network g as a decoder, such that the output of the autoencoder is $g(f(\mathbf{x}))$. We train the autoencoder to minimizing the Mean Squared Error (MSE) reconstruction loss:

$$\ell_{MSE}(\mathbf{x}, g(f(\mathbf{x}))) = \|\mathbf{x} - g(f(\mathbf{x}))\|_2^2 \,. \qquad (3)$$

The main advantage of using autoencoders is that they are unsupervised methods so no labeling is required at training time. In our case this is very relevant as we can exploit the

Algorithm 1 Image Retrieval Process

```
Require: Embedding of the dataset $f(\mathcal{D})$, query image $\mathbf{q}$, embedding $f : \mathcal{D} \longrightarrow \mathbb{R}^M$, integer
    $K \in \mathbb{N}$
Ensure: $\mathcal{R} \subseteq \mathcal{D}$ containing the first $K$ retrieved images from query $\mathbf{q}$
 1: Compute feature vector of the query $f(\mathbf{q}) \in \mathbb{R}^M$
 2: Select from $f(\mathcal{D})$ the $\widetilde{K}$ nearest neighbors $f(\mathbf{x}_i) \in \mathbb{R}^M$ to $f(\mathbf{q})$, for $i = 1, \dots, \widetilde{K}$
 3: Compute the average $\widetilde{f(\mathbf{q})}$ of $f(\mathbf{q})$ with the $\widetilde{K}$ selected points
 4: Select from $f(\mathcal{D})$ the $K$ nearest neighbors to $\widetilde{f(\mathbf{q})}$ (and store the corresponding images in $\mathcal{R}$)
```

entire dataset $\mathcal{D}$ to train our model, and in particular all the unlabelled samples in $\mathcal{D}\setminus\mathcal{T}$ that represent the largest part of IMAGO and that include images belonging to unknown classes in $\mathcal{U}$.

5.2 Image Retrieval

To perform image retrieval, we use the learned network f to preliminary compute the feature vectors for all the image in the dataset $\mathcal{D}$. This computation has to be performed only once, since we store all the computed feature vectors in the set $f(\mathcal{D}) = \{f(\mathbf{x}),\ \mathbf{x} \in \mathcal{D}\}$ to enable efficient searches in the dataset. The proposed retrieval procedure is represented in Figure 4 and detailed in Algorithm 1, and is meant to retrieve a desired number of images K. Given a query image $\mathbf{q}$, we compute the embedding $f(\mathbf{q})$ (line 1) using the trained network f, and in principle we might simply retrieve the K images whose feature vectors are the closest to $f(\mathbf{q})$.

However, to improve the quality of retrieved images with respect to a straightforward K-nearest neighbor search, we perform *query expansion*, which is shown in [1] to improve the quality of this retrieved images. In practice, we select the $\widetilde{K}$ closest feature vectors to $f(\mathbf{q})$ among all feature vectors in $f(\mathcal{D})$. Typically $\widetilde{K}$ is very small [1] and we set $\widetilde{K} = 5$. Then, the retrieved $\widetilde{K}$ feature vectors $f(\mathbf{x}_i) \in \mathbb{R}^M,\ i = 1, \dots, \widetilde{K}$ (line 2) are used to compute the *expanded* feature vector $\widetilde{f(\mathbf{q})}$ as (line 3) as

$$\widetilde{f(\mathbf{q})} = \frac{1}{\widetilde{K}+1}\left(f(\mathbf{q}) + \sum_{i=1}^{\widetilde{K}} f(\mathbf{x}_i) \right) . \tag{4}$$

In practice, the expanded feature vector $\widetilde{f(\mathbf{q})}$ is obtained by averaging $f(\mathbf{q})$ with the $\widetilde{K}$ feature vectors of the selected samples. The expanded feature vector $\widetilde{f(\mathbf{q})}$ is in practice better representative of the features that characterize the class of $\mathbf{q}$, while being less biased by the individual features of the original query image. As shown in Figure 4, we finally select the K closest feature vectors to $\widetilde{f(\mathbf{q})}$ in $f(\mathcal{D})$ via K-NN, and consider the corresponding images in $\mathcal{D}$ as the retrieved images from query $\mathbf{q}$ (line 4).

5.3 Implementation Details

We adopt a pretrained VGG16 [17] as backbone architecture, where we replace the top fully connected layers with a global averaging pooling layer resulting in an output feature vectors of $M = 512$ components, which we considered a suitable dimension for our embedding. Moreover, we add a normalization layer on top of the global averaging pooling, to set the output feature vector to have zero mean and standard deviation equals to one. This normalization improves the similarity search based on the Euclidean distance between feature vectors.

Before starting the training procedure illustrated in Section 5.1 over the modified VGG16 network, we perform a preliminary training on the labeled dataset $\mathcal{T}$ to perform classification on known classes in $\mathcal{S}$. This training allows to fine tune of the pretrained of the VGG16 architecture on the images in IMAGO dataset. Then, we discard the last dense layer and use the resulting network f to train the siamese network and the autoencoder. The decoder g is designed as the inverse of f, and its architecture is inspired to the symmetric VGG16 architecture, where we replace the pooling layers with upsampling layers. A customary procedure to enable g to reconstruct the input is to attach g after the last convolutional layer of f. Thus, the decoder g is not taking as input the feature vector extracted from f, but we nevertheless refer to the autoencoder as $g(f(\cdot))$ for the notation sake.

Finally, in all out training steps we adopt a data augmentation procedure that includes random shift, horizontal and vertical flips and change of scales. Moreover, we use the ADAM optimizer with default hyperparameters value and a batch size of 32. As a preprocessing, all the images of $\mathcal{D}$ are resized to the resolution of 224×224 and pixel values are normalized by zero-centering each channel with respect to the ImageNet data set, as described in [17].

6 Experiments

To prove the effectiveness of our solution, we compare it against three state-of-the-art image retrieval solutions in three different settings. In Section 6.4 we assess retrieval performance over the five annotated classes in $\mathcal{S}$, and show that our solution always achieves the best performance. Then, in Section 6.5 we implement a leave-one-class-out procedure to assess the retrieval performance when querying unknown classes, and we show that results are in line with the retrieval over known classes, thanks to the reconstruction loss. Finally, in Section 6.6 we investigate the ability of methods to retrieve images of the same class at different scales, where we show that considering also siamese loss is beneficial for this task.

6.1 Dataset

All our experiments are conducted on a dataset $\mathcal{D}$ composed of 35000 images, both both labelled and unlabelled, extracted from IMAGO. We exclude from $\mathcal{D}$ a test set $\mathcal{Q}$ containing 368 labelled query images uniformly distributed among the five classes in $\mathcal{S}$. To assess the performance of the considered methods, we primarily perform retrieval

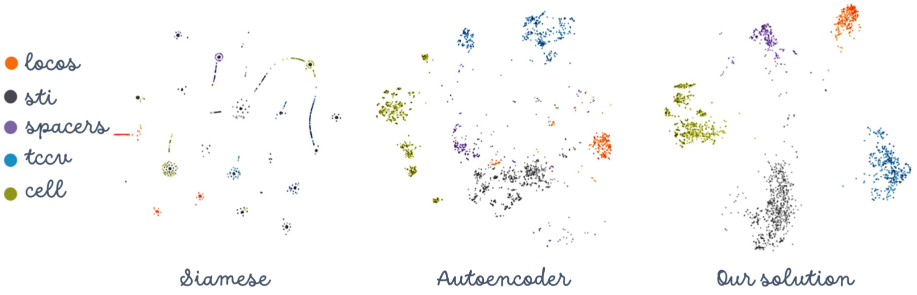


Fig. 5: Visual representation of the embedding of the labelled dataset $\mathcal{T}$ after performing a t-SNE on on feature vectors obtained respectively by siamese, autoencoder and our solution and reducing their size from M to 2.

over the labelled images in $\mathcal{T}$ (containing about 2500 images). However, to assess how our solution behaves on the entire IMAGO dataset, including on the unknown classes $\mathcal{U}$, we perform retrieval also on the whole $\mathcal{D}$ and ask experts to assess the performance, since $\mathcal{D}$ is only partially labelled.

6.2 Alternative Methods

We compare our solution against the following state-of-the-art approaches:

- **Fine-tuned VGG16**: the fine-tuned model obtained after the preliminary training described in Section 5.3 on the dataset $\mathcal{T}$ to perform classification on classes in $\mathcal{S}$.
- **Siamese**: a siamese network trained to minimize a triplet loss over semi-hard triplets sampled from $\mathcal{T}$.
- **Autoencoder**: an autoencoder based on the VGG16 architecture, trained on the dataset $\mathcal{D}$ minimizing the reconstruction loss.

We remove from $\mathcal{T}$ a small validation set that is employed to perform early stopping during training. We remark that in all the solutions the retrieval is performed as explained in Section 5.2 and under equal conditions, employing the same queries $\mathcal{Q}$ and the same dataset to retrieve samples. Due to the supervised nature of VGG16 and siamese models, in these two cases the embedding f is trained employing only the labelled portion $\mathcal{T}$ of the dataset $\mathcal{D}$.

6.3 Figures of Merit

To assess the results of retrieval methods we consider *precision* and *mean average precision*, which are computed as follows. Given a query $\mathbf{q} \in \mathcal{Q}$ and the corresponding K retrieved images, the *precision* is the fraction of correctly retrieved samples. We report precision averaged over the entire query set $\mathcal{Q}$. The *mean average precision at* K (MAP_K) takes into account how many correctly retrieved images are among the first returned results rather than the last, and is defined as in [4]. In particular, we define

	Precision at $K = 100$	$MAP_K(\mathcal{Q})$ at $K = 100$
autoencoder	0.937	0.924
fine-tuned VGG16	0.930	0.918
our	**0.986**	**0.983**
siamese	0.962	0.950

Table 1: Precision and Mean Average Precision of $\mathcal{Q}$ at $K = 100$ computed on $\mathcal{T}$.

$P(k, \mathbf{q})$ as the fraction of correctly retrieved samples among the first k images selected by the retrieval method, for $k = 1, \dots, K$. Then, the mean average precision is computed as:

$$MAP_K(\mathcal{Q}) = \frac{\sum_{i=1}^{n}(AvgP(K, \mathbf{q}_i))}{n} \tag{5}$$

where $AvgP(K, \mathbf{q}_i)$ denotes the average precision at K for the query $\mathbf{q}_i$ and is computed averaging precision over all the possible $k = 1, \dots, K$:

$$AvgP(K, \mathbf{q}_i) = \frac{\sum_{k=1}^{K}(P(k, \mathbf{q}_i) \cdot \mathbb{1}\{y_k = y_i\})}{K} \tag{6}$$

being $\mathbb{1}$ the indicator function.

6.4 Image Retrieval Performance

In the first experiment we assess the effectiveness of the considered solutions over query images that belong to known classes $\mathcal{S}$. In this experiment, for each query in $\mathcal{Q}$, we retrieve $K = 100$ images from the set of labelled images $\mathcal{T}$ and report in Table 1 the precision and mean average precision $MAP_K(\mathcal{Q})$ of all the methods. These results demonstrate that our solution retrieves the largest number of correct samples and also returns them in a better rank. Keeping a high precision among the first retrieved samples is very important in our scenario, where engineers might possibly look only at the most relevant results. The second best-performing solution is the siamese network which, like ours, leverages annotations. We speculate that the performance gap between our solution and the siamese network is due to the advantage of using unsupervised images during training.

To get a visual idea of the embedding in the Euclidean space, we perform a t-SNE on feature vectors corresponding to the entire dataset $\mathcal{D}$, reducing their dimension from M to 2. Figure 5 reports the distribution of the embedded samples obtained respectively by siamese, autoencoder and our solution. It can be noticed that our solution clearly groups together vectors corresponding to images of the same class better than the other two methods, enabling better retrieval.

We also assess our solution on the whole dataset $\mathcal{D}$, which includes images from unknown classes $\mathcal{U}$. The entire query set $\mathcal{Q}$ has been tested and the retrieval results have been manually validated by experts. Since performance assessment required such a visual inspection, we were not able to consider alternative methods, and we restricted to only $K = 30$ results. Our method achieves a precision of 0.93, confirming that our solution

	no cell	no locos	no spacers	no sti	no tccv	mean
autoencoder	**0.859**	0.290	**0.518**	**0.744**	0.452	0.573
fine-tuned VGG16	0.807	**0.540**	0.218	0.534	0.314	0.482
our	0.800	0.456	0.489	0.671	**0.845**	**0.652**
siamese	0.587	0.435	0.246	0.406	0.369	0.409

Table 2: Precision at $K = 100$ of the considered methods trained using leave-one-class-out cross validation. The results are averaged only over the queries belonging to the excluded class.

	no cell	no locos	no spacers	no sti	no tccv	mean
autoencoder	**0.805**	0.101	**0.362**	**0.631**	0.270	0.434
fine-tuned VGG16	0.749	**0.444**	0.110	0.447	0.237	0.398
our	0.735	0.255	0.344	0.556	**0.773**	**0.533**
siamese	0.446	0.319	0.160	0.238	0.259	0.284

Table 3: Mean average precision at $K = 100$ for the cross validation leave on out scenario on the considered models. The results are averaged only over the queries belonging to the excluded class.

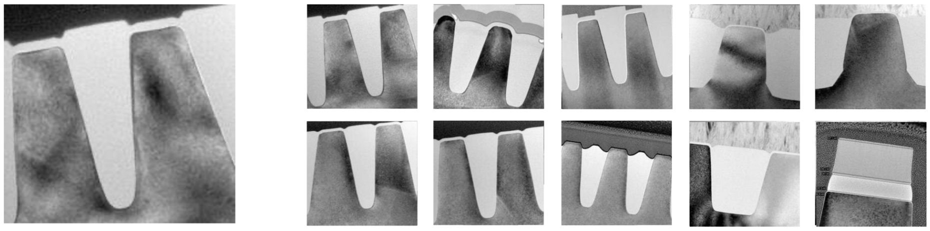
is very effective even in the more realistic scenario where the retrieval is performed over a very large set including images from unknown classes.

6.5 Retrieval of Images from Unknown Classes

In the second experiment we evaluate the considered methods in retrieving query images belonging to unknown classes $\mathcal{U}$ not represented in the training set $\mathcal{T}$. To this purpose, we perform leave-one-out cross validation, excluding every time all the labelled training samples belonging to a specific class. We then test the trained models on query images belonging to the excluded class only. Observe that none of the considered solutions were trained with label information from the excluded images, and the only on semantic information comes from other classes. It is worth however remarking that methods using unlabelled samples might have access to instances of the excluded class that appears unlabelled in $\mathcal{D}$. We again compare all the methods in terms of precision and mean average precision at K, and the results are reported in Tables 2 and 3, respectively. Each column corresponds to an excluded class, and the figures of merit are averaged over query images belonging to the excluded class only. The last column reports the corresponding figure of merit averaged over all the excluded classes. In this scenario, we need to take into account that the siamese network and the fine-tuned VGG16 have never seen a sample from the excluded class during training, since their training completely relies on labelled samples. Not surprisingly, the best performing solutions are those leveraging unsupervised loss, namely the autoencoder and our solution, which however cannot take a relevant advantage from the siamese loss during training.

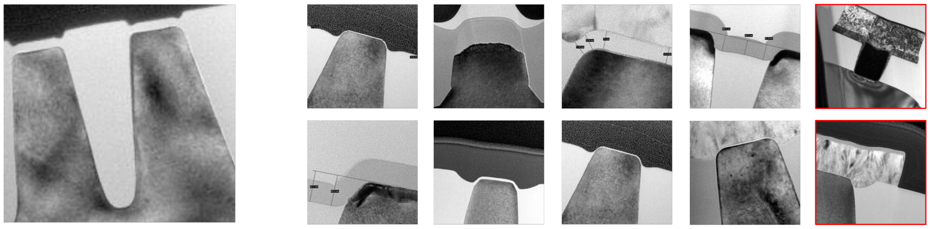
6.6 Retrieval of Images at Different Scales

The last experiment is designed to assess the performance of the methods in retrieving images of the same class that appear very different due to the magnification levels at



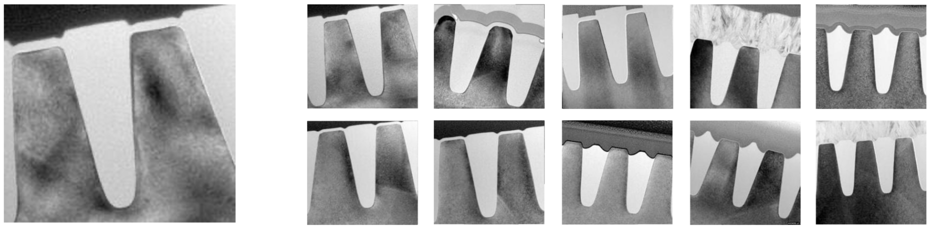
Query Retrieved images (our)

Fig. 6: Example of retrieval performed by our solution. Image on the left is the query, belonging to the sti class, while on the right the first ten retrieved images are reported. Our solution performs a successful retrieval, selecting images belonging to the same class of the query, and including images taken at various magnification levels.



Query Retrieved images (siamese)

Fig. 7: Example of retrieval performed by the siamese solution. The image on the left is the query, belonging to the sti class, while on the right the first ten retrieved images are reported. Differently from our solution, the siamese solution retrieves two images belonging to wrong classes (circled in red) while most of the correct ones where taken at different magnification levels than the query.



Query Retrieved images (autoencoder)

Fig. 8: Example of retrieval performed by the autoencoder solution. Image on the left is the query, belonging to the sti class, while on the right the first ten retrieved images are reported. Differently from our solution, the autoencoder does not retrieve images taken at different magnification levels than the query.

which images were acquired. It is in fact very important, according to STMicroelectronics engineers, for the system to return images selected at different scales. To this purpose, we measure the variability of the scales among the retrieved images.

Examples of retrievals are reported for a qualitative analysis in Figures 6, 7 and 8. Image on the left is the query, belonging to the class representing the sti structure, while on the right the first ten retrieved images by each method are reported. Figure 6 shows

	no cell	no locos	no spacer	no sti	no tccv	mean
autoencoder	2.073	3.167	3.358	3.099	3.466	3.033
fine-tuned VGG16	1.804	1.863	2.639	2.700	1.907	2.182
our	1.817	2.667	4.001	4.210	3.193	3.178
siamese	2.456	2.940	3.058	4.483	3.198	3.227

Table 4: Standard deviation of the scale of the correctly retrieved images using all methods (the higher, the better).The results are averaged only over the queries belonging to the excluded class.

that our solution successfully retrieves only images belonging to the correct class, and that it includes images taken at different magnification levels than the query. On the contrary, the other approaches fail either in retrieving the correct class (siamese model, Figure 7), or in addressing the semantic similarity between images, only retrieving samples taken at the same magnification level than the query (autoencoder, Figure 8).

To quantitatively assess the ability of the methods in retrieving images at different scales, we report in Table 4 the standard deviation of scale values computed from the correctly retrieved images, which is expected to be large when the model can retrieve images at different scales. As expected, the method that retrieves images with a wider range of scales is the siamese model, since it is only fed with labelled samples. We speculate that training f as a siamese architecture pushes the network towards identifying specific semantic features of the images, which might appear at different scales. In contrast, the reconstruction loss can only rely on the visual similarity of the data. This perhaps prevents the autoencoder from retrieving images at different scales. Overall, the results confirm that our method is able to retrieve images with a significant scale variety, effectively integrating both the peculiarities of siamese networks and autoencoders.

7 Conclusions

In this paper we address image retrieval in the high-tech scenario of semiconductor manufacturing. The proposed solution consists in a new training procedure for deep neural networks, which alternates the optimization of two losses: a triplet loss on annotated samples and the reconstruction loss on all the samples disregarding whether they are annotated or not. Our experiments demonstrate that our solution outperforms state-of-the-art alternatives on the IMAGO dataset, which contains two millions of TEM images acquired at STMicroelectronics in Agrate Brianza, Italy. Even though the training procedure was designed to cope with the peculiarities of the IMAGO dataset, which is only partially labelled and where annotations cover only a fraction of the total number of classes, our solution is very general, and can be easily applied to a wide variety of data. Ongoing work on the subject includes testing the method on a broader portion of the annotated dataset IMAGO and design training procedure to further strengthen the network invariance to the severe scale changes characterizing these images.

References

1. Ahmed, A., Malebary, S.J.: Query expansion based on top-ranked images for content-based medical image retrieval. IEEE Access **8**, 194541–194550 (2020). https://doi.org/10.1109/ACCESS.2020.3033504
2. Balmachnova, E., Florack, L., Haar Romeny, B.t.: Feature vector similarity based on local structure. In: International Conference on Scale Space and Variational Methods in Computer Vision. pp. 386–393. Springer (2007)
3. Chung, Y., Weng, W.: Learning deep representations of medical images using siamese cnns with application to content-based image retrieval. CoRR **abs/1711.08490** (2017), http://arxiv.org/abs/1711.08490
4. Cooper, W.S.: Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. American documentation **19**(1), 30–41 (1968)
5. Duan, J., Kuo, C.C.J.: Bridging gap between image pixels and semantics via supervision: A survey. arXiv preprint arXiv:2107.13757 (2021)
6. Dubey, S.R.: A decade survey of content based image retrieval using deep learning. IEEE Transactions on Circuits and Systems for Video Technology pp. 1–1 (2021). https://doi.org/10.1109/TCSVT.2021.3080920
7. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International workshop on similarity-based pattern recognition. pp. 84–92. Springer (2015)
8. Kan, S., Cen, Y., He, Z., Zhang, Z., Zhang, L., Wang, Y.: Supervised deep feature embedding with handcrafted feature. IEEE Transactions on Image Processing **28**(12), 5809–5823 (2019)
9. Kato, T.: Database architecture for content-based image retrieval. In: Jamberdino, A.A., Niblack, C.W. (eds.) Image Storage and Retrieval Systems. vol. 1662, pp. 112 – 123. International Society for Optics and Photonics, SPIE (1992), https://doi.org/10.1117/12.58497
10. Kaya, M., Bilge, H.Ş.: Deep metric learning: A survey. Symmetry **11**(9), 1066 (2019)
11. Krizhevsky, A., Hinton, G.E.: Using very deep autoencoders for content-based image retrieval. In: ESANN. vol. 1, p. 2. Citeseer (2011)
12. Pandey, A., Mishra, A., Verma, V.K., Mittal, A., Murthy, H.: Stacked adversarial network for zero-shot sketch based image retrieval. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2540–2549 (2020)
13. Qi, Y., Song, Y.Z., Zhang, H., Liu, J.: Sketch-based image retrieval via siamese convolutional neural network. In: 2016 IEEE international conference on image processing (ICIP). pp. 2460–2464. IEEE (2016)
14. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
15. Shen, Y., Qin, J., Chen, J., Yu, M., Liu, L., Zhu, F., Shen, F., Shao, L.: Auto-encoding twin-bottleneck hashing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2818–2827 (2020)
16. Sheng, H., Zheng, Y., Ke, W., Yu, D., Cheng, X., Lyu, W., Xiong, Z.: Mining hard samples globally and efficiently for person reidentification. IEEE Internet of Things Journal **7**(10), 9611–9622 (2020)
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)
18. Siradjuddin, I.A., Wardana, W.A., Sophan, M.K.: Feature extraction using self-supervised convolutional autoencoder for content based image retrieval. In: 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS). pp. 1–5 (2019). https://doi.org/10.1109/ICICoS48119.2019.8982468

19. Smeulders, A., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(12), 1349–1380 (2000). https://doi.org/10.1109/34.895972
20. Wang, Y., Ou, X., Liang, J., Sun, Z.: Deep semantic reconstruction hashing for similarity retrieval. IEEE Transactions on Circuits and Systems for Video Technology **31**(1), 387–400 (2020)