International Journal of Neural Systems, Vol. 0, No. 0 (2005) 1–15

A conditional Generative Adversarial Network and transfer learning-oriented anomaly classification system for electrospun nanofibers

COSIMO IERACITANO
DICEAM, University Mediterranea of Reggio Calabria,
Via Graziella Feo di Vito,
Reggio Calabria, 89124, Italy
E-mail: cosimo.ieracitano@unirc.it

NADIA MAMMONE
DICEAM, University Mediterranea of Reggio Calabria,
Via Graziella Feo di Vito,
Reggio Calabria, 89124, Italy
E-mail: nadia.mammone@unirc.it

ANNUNZIATA PAVIGLIANITI
Polytechnic University of Turin,
Corso Castelfidardo,
Turin, 10129, Italy
E-mail: annunziata.paviglianiti@polito.it

FRANCESCO CARLO MORABITO
DICEAM, University Mediterranea of Reggio Calabria,
Via Graziella Feo di Vito,
Reggio Calabria, 89124, Italy
E-mail: morabito@unirc.it

In this study, a generative model and transfer learning powered system for classification of Scanning Electron Microscope (SEM) images of defective nanofibers (D-NF) and non-defective nanofibers (ND-NF) produced by electrospinning process is proposed. Specifically, a conditional-Generative Adversarial Network (c-GAN) is developed to generate synthetic D-NF/ND-NF SEM images. A *transfer learning-oriented* strategy is also proposed. First, a Convolutional Neural Network (CNN) is pre-trained on real images. The *transfer-learned CNN* is trained on synthetic SEM images and validated on real ones, reporting accuracy rate up to 95.31%. The achieved encouraging results endorse the use of the proposed generative model in industrial applications as it could reduce the number of needed laboratory electrospinning experiments that are costly and time consuming.

Keywords: Deep Learning; Generative Adversarial Networks; Transfer Learning; Convolutional Neural Networks; Electrospinning; Nanomaterials; Classification.

1. Introduction

Electrospinning (ES) is one of the most versatile and viable techniques due to its easiness and cost-effectiveness, employed to generate polymeric ultra-thin fibers with diameters in the nanoscale range, known indeed as *nanofibers* (NF).[1] In recent years, electrospun NF have been gaining attention in different areas of nanotechnology including sensors,

electronics, tissue engineering, drug delivery, filters, wound dressing or functional garments.[2] However, NFs can be affected by structural anomalies (defects) that appear as beaded structures along the electrospun network. Defective nanofibers (D-NF) prevent the use of such materials in any nanotechnology application. Hence, monitoring the quality of NF produced by the electrospinning process is of great importance especially in the industrial manufacturing. The Scanning Electron Microscope (SEM) is usually employed to inspect the electrospun NF material. This procedure requires a high time-effort from expert operators. Moreover, visual examination of SEM images is not an effective method to distinguish non-defective NF (ND-NF) from D-NF. Automatic classification systems would lead to a significant acceleration of the production chain.

In recent years there has been an exceptional use of Artificial Intelligence (AI) in many disciplines (e.g, health care,[3] sentiment analysis,[4] image recognition[5]). Several AI-based classification systems have been developed for a wide variety of fundamental applications, achieving impressive performance, similar or even better than the human decision. The application of AI in nanotechnology has been also highly increasing over the recent years.[6,7] Nevertheless, there is a limited numbers of studies concerning AI for nanomaterials produced by electrospinning. Some studies focused on the automatic detection and localization of defects in SEM images. Boracchi et al.[8] proposed a sparse-based representation technique for anomaly detection in SEM images. Carrera et al.[9] developed a dictionary learning-based framework of anomaly-free sub-patches to identify defects in SEM images, reporting an Area Under the ROC curve (AUROC) of 92.6%, over a dataset of 45 NF SEM images (40 with anomalies and 5 anomaly-free), outperforming other sparse and similarity based-methods.[10,11] Napoletano et al.[12] proposed a region-based approach to recognize and pinpoint morphological anomalies in SEM images by means of a Convolutional Neural Network (CNN) and self-similarity techniques. The method outperformed Carrera et al.[9] approach, achieving AUROC of 97%, using the same SEM image dataset. Other works, instead, focused on the automatic classification of SEM images of D-NF and ND-NF. In this context, there are few works in the current state-of-the-art. Specifically, Ieracitano et al.[13] developed a deep CNN to classify raw SEM images of D-NF and ND-NF, trained and tested over an experimental small dataset of 160 images (85 D-NF and 75 ND-NF), reporting classification accuracy rate up to 80%; while, in[14] the authors proposed an optimized deep CNN based on image preprocessing by means of Sobel filtering, achieving average accuracy of 80.27±4.8%.

It is to be noted that one of the main drawbacks of the above studies was the limited number of the available SEM images. Indeed, a massive amount of training samples is necessary to effectively train deep learning (DL) models and avoid overfitting issues. However, collecting new SEM images requires new costly and time-consuming laboratory experiments. So, the application of *data augmentation* techniques would be of great help to address this problem. Recently, augmentation strategies have been proposed. In[15] a hybrid unsupervised-supervised machine learning system was presented and a relatively naive data augmentation technique was employed by corrupting the features extracted from the proposed Autoencoder (AE) with white Gaussian noise; whereas, in[16] the size of the dataset[13] was increased by means of common augmentation method (noise, translations and rotations). However, standard augmentation approaches are based on the use of the same, slightly changed, images. Hence, motivated by the need of new methodologies able to generate brand new SEM images to reduce the number of costly experiments and resources, here, a Generative Adversarial Networks (GAN) and transfer learning based strategy is proposed. In particulr, GAN have been widely used to create high quality realistic images (i.e., synthetic samples) in several applications.[17] In this study, in order to cope with the issue of the limited size of the dataset and of labeled samples, a GAN-based approach is proposed for synthetic SEM image generation. To the best of our knowledge this is the first work that uses adversarial networks to produce synthetic SEM images of electrospun nanofibers.

A conditional-GAN (c-GAN) is developed to capture real features from original data and generate *synthetic* SEM images of D-NF and ND-NF. First, a customised CNN is trained and tested on real data (here referred to as *pre-trained CNN*). Second, synthetic samples are used as input to the *pre-trained CNN* according to a *transfer learning-oriented* strategy.[18] Finally, in order to prove the effectiveness of

the "fake" images generated by the proposed c-GAN, the *transfer-learned CNN* is subsequently tested on real samples. The major contributions of this paper are summarized as follows:

- generation of realistic-looking SEM images of defective and non-defective NF by means of a c-GAN, thus reducing the need of costly laboratory tests.
- development of a *transfer learning-oriented* classification system (i.e., *transfer-learned CNN*) based on synthetic high-quality images.
- development of a system with a high potential for industrial deployment in nanotechnology applications.

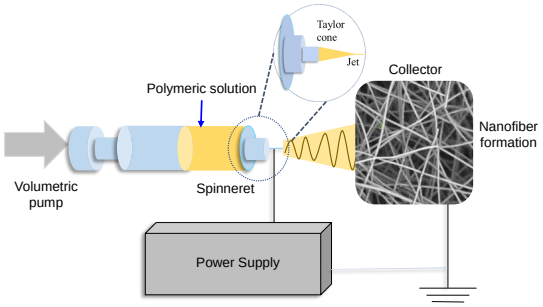
The rest of this work is organized as follows. Section 2 introduces the electrospinning process, the dataset description and the proposed methodology, including: the development of the c-GAN for *synthetic*-SEM image generation, the *transfer learning-oriented* classification system and performance metrics. Section 3 reports the classification performance. Section 4 discusses the achieved results, while Section 5 concludes the paper.

2. Materials and Methodology

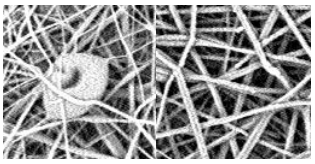
The proposed framework comprises the following processing stages: (i) electrospinning process and generation of SEM images dataset; (ii) generation of *synthetic*-SEM images of electrospun nanofibers by means of the proposed c-GAN; (iii) pre-training of a custom CNN on *real*-SEM images (*pre-trained CNN*); (iv) transfer learning and SEM images classification (*transfer-learned CNN*).

2.1. *Electrospinning process*

The electrospinning process is one of the most efficient techniques to produce fibers with micro-or-nano-diameters.[19] A common electrospinning apparatus consists of a high voltage power supply, a spinneret, a volumetric pump and a collector, as reported in Fig. 1a. A polymer solution is contained in a syringe and fed via a spinneret using a volumetric pump at a constant flow rate. A pendant droplet is extruded from the spinneret. A high voltage power supply is applied to generate an electric field between the needle tip of the syringe and the collector. At a particular voltage the electrostatic force overcomes the surface tension of the polymer solution and the droplet deforms into a Taylor cone, from which a fluid jet is further extruded towards the collector. During the excursion of the jet from the needle tip to the collector, the solvent evaporates, and narrow solid fibers (with micro or nanodiameter) are deposited on the grounded collector surface with random orientation.



(a)



(b) (c)

Figure 1: (a) Electrospinning apparatus. (b) Example of defective nanofibers SEM image. (c) Example of non-defective nanofibers SEM image.

2.1.1. *Applications of Electrospun Nanofibers*

Electrospun nanofibers are used in several real-world applications, such as biomedical, environment and energy storage fields. For example, in the biomedical field, mechanical and biodegradable properties of nanomaterials are exploited to repair or regenerate damaged tissues or organs.[20] NFs are are also employed in drug delivery applications since they are able to maintain the integrity and bioactivity of the therapeutic agent.[21] In the environmental field, NFs are used as advanced filters for the removal of pollutants such as toxic ions and organic molecules from both polluted air and wastewater.[22] Finally, in the energy field, NFs can be employed for the development of electrodes for rechargeable solar cells batteries, supercapacitors, and fuel cells, since they have very good conductivities and structural stability.[22]

2.1.2. *Anomalies in Nanofibers*

Nanofibers are often characterized by the formation of undesired defects known as *beads*. Electrospun beaded nanofibers are caused by the instability of the polymeric jet solution. Electrospinning parameters (such as concentration of the solution, surface tension, electric potential, flow rate, tip-to-collector distance) can influence the production of NFs, leading to the possible presence of anomalies.[23] For example, at concentrations too low, electrospray occurs instead of electrospinning, causing the formation of beads of spherical shape. As concentration increases, the bead modifies its shape from spherical to a spindle-like one. Increasing the applied voltage promotes a shrinking of the fiber diameter up to the formation of anomalies.[24] The flow rate of the polymer is a parameter that defines the transfer rate of the material. If the forward speed is too fast it induces the formation of defects.[25] Finally, too near or too far distance from the tip-collector is a cause of beads production.[26]

Fig. 1b shows an example of bead formation during electrospinning (i.e., defective nanofibers, D-NF); while Fig. 1c an example of anomaly-free nanofiber (i.e, non-defective nanofibers, ND-NF).

2.1.3. *Dataset description*

The experimental activities were carried out at the *Materials for Environmental and Energy Sustainability Laboratory*, University "Mediterranea" of Reggio Calabria (Italy). The Polyvinylacetate (PVAc) was used as polymer and Ethanol (EtOH) as solvent. The morphology of the electrospun nanofibers were then investigated using the Phenom Pro-X Scanning Electron Microscope (SEM) and the Fibermetric analyzer to observe the NFs structure and the possible presence of undesired defects (i.e., beads). During the experiments the four main control electrospinning parameters, namely, concentration (c_1), applied voltage (c_2), flow rate (c_3) and tip-to-collector distance (TCD, c_4), have been changed in the well-known working ranges (c_1 [10;25] $\%wt$; c_2 [10;17.5] kV ; c_3 [100;300] $\mu L/min$; c_4 [10-15] cm). Seventeen electrospinning experiments were conducted. In particular, the set-up of each experiment is reported in Table 1. Each sample of the electrospun nanofibers was analyzed through the SEM and 47 representative areas were selected by an expert operator. Overall, a dataset of 17 x 47=799 SEM images sized 128 x 128 was generated, according the procedure reported in.[13] However, the proposed c-GAN was not able to model 128 x 128 images, possibly due to the limited number of samples together with a high complexity of the images. Hence, in this study, each image was further divided into four patches sized 64 x 64 for an overall set of about 3200 SEM images (799 x 4 = 3196). This allowed to provide a massive amount of data required to train the proposed c-GAN.

2.1.4. *Dataset preparation*

The experimental dataset (here denoted to as DB_{all}) consisted of 3196 SEM images (∼3200): 1536 belonging to D-NF class, 1660 belonging to ND-NF class. Since training generative networks requires huge amounts of data, 80% of the SEM images dataset was allocated for training the proposed c-GAN. This subset, here denoted to as $DB_{80\%}$, consisted of 2560 SEM images: 1230 D-NF and 1330 ND-NF. $DB_{80\%}$ is further randomly partitioned according to a k-fold cross-validation (with k=10) approach for developing the custom *pre-trained CNN*. Hence, 256 images (123 D-NF and 133 ND-NF) were iteratively used as test set and the remaining 2304 (1107 D-NF and 1197 ND-NF) as train set. The remaining 20% of data was used to test the *transfer-learned CNN*. This subset is here denoted to as $DB_{20\%}$ and consisted of 640 SEM images: 308 D-NF and 332 ND-NF. Table 2 reports the dataset details and classes distribution.

Table 1: Setup of the experimental electrospinning parameters.

#	**Concentration** (c_1) [$\%wt$]	**Voltage** (c_2) [kV]	**Flow rate** (c_3) [$\mu L/min$]	**TCD** (c_4) [cm]
1	10	15	100	10
2	15	10	100	10
3	15	13.5	100	10
4	15	15	100	10
5	15	15	200	10
6	15	15	300	10
7	15	15	100	12.5
8	15	15	100	13.5
9	15	15	100	15
10	18	15	100	15
11	20	10	100	10
12	20	11.5	100	10
13	20	13.5	100	10
14	20	15	100	10
15	20	16	100	10
16	20	17.5	100	10
17	25	15	100	10

Table 2: Dataset and classes distribution of the *real*-SEM images. Note that $DB_{80\%}$ is used to train the proposed c-GAN and *pre-trained CNN*; while, $DB_{20\%}$ is used to test the proposed *transfer-learned CNN*.

Class	**DB**$_{all}$	**DB**$_{80\%}$	**DB**$_{20\%}$
D-NF	1536	1230	308
ND-NF	1660	1330	332
Total	3196	2560	640

2.2. *Generation of synthetic SEM electrospun nanofibers' images*

2.2.1. *Conditional Generative Adversarial Network*

A GAN consists of two separate neural networks called *generator* (G) and *discriminator* (D), respectively, trained adversarially. G is trained to generate synthetic samples and fool the discriminator; whereas, D is trained to distinguish between real and fake data. More specifically, the generator learns a distribution p_g over the data x by mapping a random input noise distribution $p_z(z)$ to the data space as $G(z, \theta_g)$, where θ_g are the parameters of the generative network; on the other hand, the discriminator is defined as $D(x, \theta_d)$, where θ_d are the parameters of the discriminatory network outputs a single scalar $D(x)$ and refers to the probability that x belongs to real data rather than generated by G. It is to be noted that the generator and the discriminator are trained simultaneously, according to the *two-player min-max game* logic and value function $V(G, D)$:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[log(1-D(G(z)))] \quad (1)$$

D is trained in order to maximize the probability to assign the correct label (i.e., real or synthetic) while G is trained to minimize $log(1 - D(G(z)))$. Further details on this procedure are reported in.[27]

A conditional generative adversarial network is an extension of a standard GAN where both generator and discriminator are conditioned on an additional information y, i.e., the class label.[28] In particular, the label y feeds the generator in order to produce samples corresponding to the class y; and feeds the discriminator to improve the real data recognition. The objective loss function for conditional-GAN is similar to eq. 2:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[logD(x|y)]+E_{z\sim p_z(z)}[log(1-D(G(z|y)))] \quad (2)$$

2.2.2. *Proposed conditional-GAN architecture*

The proposed c-GAN, including the specific architectures of the generator and discriminator, are depicted in Fig. 2(a) and 2(b), respectively, and are based on CNNs,[29] while Table 3 reports the architecture details in tabular format.

Generator G. The random noise vector (sized 1 x 100) is firstly projected and reshaped into 4 x 4 x 512 arrays. The categorical label is integrated by passing it through an embedding layer that maps it into a single vector, later passed through a fully connected layer. In this study, the output is reshaped into a unique feature map 4 x 4. This is to match the 4 x 4 x 512 activations of the first layer. Indeed, the new 4 x 4 map is concatenated to the existing 512, resulting in a 4 x 4 x 513 input matrix. The latter is further up-sampled to 64 x 64 (i.e., size of the real D-NF/ND-NF SEM images) by means of a series of transposed convolution layers (*tconv*) followed by batch normalization and ReLU layers. In this study, the generator includes 4 transposed convolution layers ($tconv_i$, with $i = 1, .., 4$) with filter size 5 x 5, stride of 1 for $tconv_1$, stride of 2 for $tconv_{2,3,4}$, and a number of filters that decreases in each layer.

Discriminator (D). The discriminator takes as input the generated SEM image obtained from G and a real one, with the aim to infer if the generated SEM image is real or not. Both images have the same dimension that in this study is 64 x 64. As in the generator, the class label is passed through an embedding and reshape layer. The output is reshaped into a single 64 $\times$ 64 activation map and concatenated with the input image, resulting in a 64 x 64 x 2 input matrix. D consists of a series of convolution layers (*conv*) with batch normalization and leaky ReLU layers (Fig. 2(b)). In this study, the discriminator includes 5 convolutional layers ($conv_i$, with $i = 1, .., 5$) with 64, 128, 256, 512 and 1 filters, respectively. $conv_{1-4}$, have filters 5 x 5 and stride 2,

while $conv_5$ has filter 4 x 4 and stride 1.

It is worth mentioning that the topology of the proposed c-GAN was selected following the DCGAN architecture reported in Radford et al.[30] and according to a trial-and-error approach. Indeed, Table 4 reports different architectures developed by changing the number of filters and processing layers: c-GAN$_1$, c-GAN$_2$, c-GAN$_3$. The generative models were trained for about 400 epochs and learning rate of 10^{-3}. Note that the c-GAN with the lowest Fréchet inception distance (FID) score[31] (i.e., c-GAN$_3$) was selected as the best model. Furthermore, for a comparative analysis, different bunches of realistic looking (i.e., *synthetic*) SEM images are generated by c-GAN$_3$ (here merely denoted as c-GAN). In particular, in this study, three sets of synthetic data (DB_s) of size 500, 1600, 3196 (i.e., same size of the real SEM images dataset) have been produced and used to train the proposed *transfer-learned CNN*. In addition, for fair comparisons, the same class distribution of the real dataset was reproduced, i.e., 48% of data was generated as D-NF and 52% as ND-NF. Details are reported in Table 5.

2.3. *Transfer learning powered CNN for SEM images classification*

Transfer learning refers to the procedure by which the knowledge (i.e., features) learned by an already trained network is exploited to improve the learning of a new set of data.[32]A pre-trained custom CNN, developed to classify real D-NF/ND-NF SEM images achieved by electrospinning experiments, is employed.

Table 3: Setup of the proposed c-GAN.

Generator	Activations	Discriminator	Activations
Input (Noise)	100	Input (Image)	64x64x1
Project and reshape	4x4x512	Embed and reshape (Label)	64x64x1
Embed and reshape (Label)	4x4x1	Concatenation	64x64x2
Concatenation	4x4x513	Convolution filters= 5x5, stride=2	32x32x64
Transposted Convolution filters= 5x5, stride=2	8x8x256	Batch Norm + Leaky ReLU	32x32x64
Batch Norm + ReLU	8x8x256	Convolution filters= 5x5, stride=2	16x16x128
Transposted Convolution filters= 5x5, stride=2	16x16x128	Batch Norm + Leaky ReLU	16x16x128
Batch Norm + ReLU	16x16x128	Convolution filters= 5x5, stride=2	8x8x256
Transposted Convolution filters= 5x5, stride=2	32x32x64	Batch Norm + Leaky ReLU	8x8x256
Batch Norm + ReLU	32x32x64	Convolution filters= 5x5, stride=2	4x4x512
Transposted Convolution filters= 5x5, stride=2	64x64x1	Batch Norm + Leaky ReLU	4x4x512
Tanh function	64x64x1	Convolution filters= 4x4, stride=2	1x1x1

2.3.1. *pre-trained CNN*

The proposed *pre-trained CNN* is reported in Fig. 3(a). It is composed of a series of *conv*, ReLU, max-pooling (*mpool*) layers and ends with a standard artificial neural network (ANN) for classification purposes. The CNN has two *conv* layers with 64 and 96 filters of size 4 x 4, stride of 2 and padding of 1, resulting in 64 features maps of size 32 x 32 and 96 features maps of size 8 x 8, respectively. After applying ReLU activation function, the extracted features maps are downsampled by means of a max pooling operation achieving maps size on 16 x 16, 4 x 4. All max pooling layers have filters sized 2 x 2 with a stride of 2. Finally, the extracted feature maps are reshaped into 1-dimensional vector 1 x D (with D = 1536) and input to a 1-hidden layer NN of 1000 neurons and a softmax output layer to perform the 2-way discrimination task: D-NF vs. ND-NF. The Adaptive Moment (ADAM) optimization technique is used for training the network for a number of epochs=250 with the following training parameters: β_1= 0.9; β_2=0.999 (with $\beta_{1,2}$ exponential decay rates); learning rate α=10^{-3}. The network was implemented in Matlab R2021b on a workstation with Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz, one NVIDIA GeForce RTX 2080 Ti GPU and 64 GB RAM. Note that the network was trained and tested using only *real*-SEM images. In particular, the DB$_{80\%}$ was employed. The k-fold cross-validation technique (with k=10) was applied, for an overall training time of about 30 min. Hence, 256 images (123 D-NF and 133 ND-NF) were iteratively used as test set and the remaining 2304 (1107 D-NF and 1197 ND-NF) as train set. The model corresponding to the best performance was used as *pre-trained CNN*.

2.3.2. *transfer-learned CNN*

The *pre-trained CNN* is used as a starting point to validate the synthetic SEM images generated by the c-GAN. In this study, the features from the early layers of the *pre-trained CNN* were kept fixed, in other words, transferred to the new model; while the last layers, namely, fully connected and classification layers, were re-trained (i.e., the weights are discarded and then trained again). The *transfer-learned CNN* model is trained on 3196 (1536 D-NF, 1660 ND-NF) synthetic SEM images (i.e., same size of the real dataset) to perform the binary D-NF vs. ND-NF

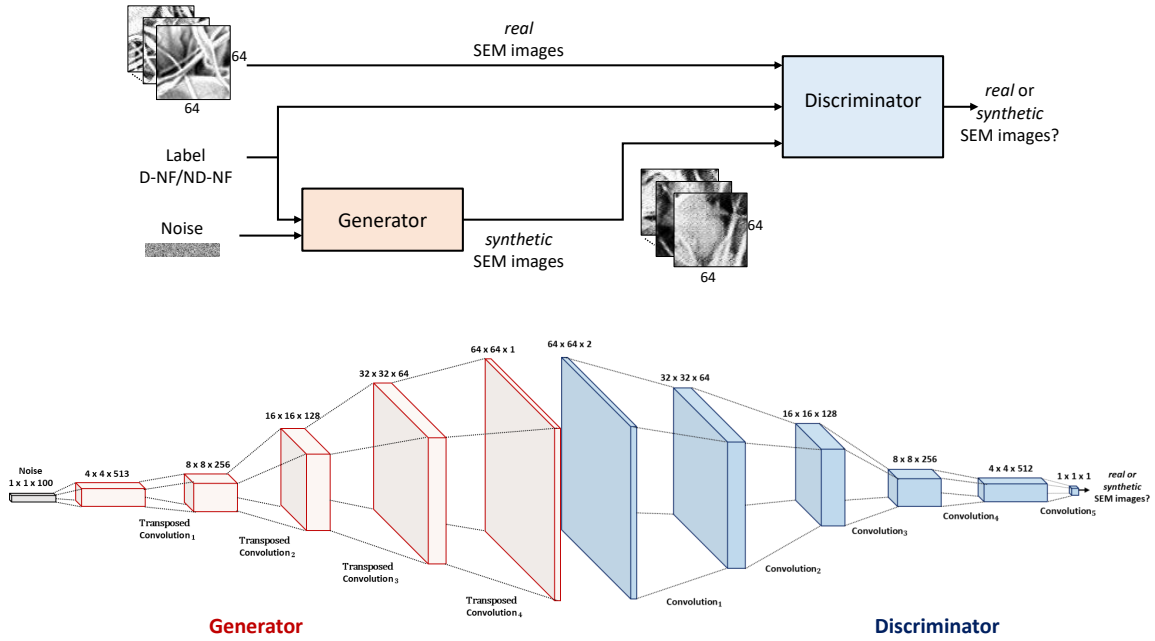


Figure 2: (a) The architecture of the proposed conditional-GAN. (b) Architecture of the generator and discriminator.

Table 4: Different conditional-GAN configurations. Note that c-GAN$_3$ refers to the c-GAN employed in this study.

Generator	**c-GAN$_1$**	**c-GAN$_2$**	**c-GAN$_3$**		**Discriminator**	**c-GAN$_1$**	**c-GAN$_2$**	**c-GAN$_3$**
Input (Noise)	100	100	100		Input (Image)	64x64x1	64x64x1	64x64x1
Project and reshape	16x16x128	8x8x256	4x4x512		Embed and reshape (Label)	64x64x1	64x64x1	64x64x1
Embed and reshape (Label)	16x16x1	8x8x1	4x4x1		Concatenation	64x64x2	64x64x2	64x64x2
Concatenation	16x16x129	8x8x257	4x4x513		Convolution filters= 5x5, stride=2	32x32x64	32x32x64	32x32x64
Transposted Convolution filters= 5x5, stride=2	32x32x64	16x16x128	8x8x256		Batch Norm + Leaky ReLU	32x32x64	32x32x64	32x32x64
Batch Norm + ReLU	32x32x64	16x16x128	8x8x256		Convolution filters= 5x5, stride=2	16x16x128	16x16x128	16x16x128
Transposted Convolution filters= 5x5, stride=2	64x64x1	32x32x64	16x16x128		Batch Norm + Leaky ReLU	-	16x16x128	16x16x128
Batch Norm + ReLU	-	32x32x64	16x16x128		Convolution filters= 5x5, stride=2	-	8x8x256	8x8x256
Transposted Convolution filters= 5x5, stride=2	-	64x64x1	32x32x64		Batch Norm + Leaky ReLU	-	8x8x256	8x8x256
Batch Norm + ReLU	-	-	32x32x64		Convolution filters= 5x5, stride=2	-	-	4x4x512
Transposted Convolution filters= 5x5, stride=2	-	-	64x64x1		Batch Norm + Leaky ReLU	-	-	4x4x512
Tanh function	64x64x1	64x64x1	64x64x1		Convolution filters= 4x4, stride=2	1x1x1	1x1x1	1x1x1

classification for about 300 epochs using the ADAM optimization technique. Finally, the *transfer-learned CNN* is tested on new *real*-SEM images that were not used for training the proposed c-GAN and *pre-trained CNN* (i.e., DB$_{20\%}$, Table 2).

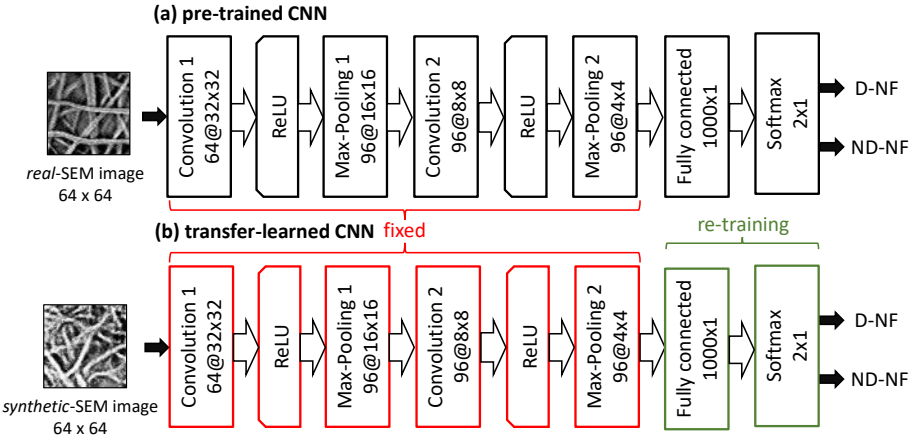


Figure 3: (a) The architecture of the *pre-trained CNN* and (b) the transfer learning approach.

2.4. *Classification metrics*

The performance of the proposed CNN were estimated by means of standard classification metrics, i.e., specificity (Sp), sensitivity (Ss), positive predicted value (PPV), negative predicted value (NPV), F1-score (FS), accuracy (Acc)), which are defined as: $Sp = \frac{TN}{TN+FP}$; $Ss = \frac{TP}{TP+FN}$; $PPV = \frac{TP}{TP+FP}$; $NPV = \frac{TN}{TN+FN}$; $FS = \frac{2*TP}{2*TP+FP+FN}$; $Acc = \frac{TP+TN}{TP+TN+FP+FN}$.
with TP (true positive) number of D-NF images classified as D-NF; TN (true negative) number of ND-NF images classified as ND-NF; FP (false positive) number of ND-NF images detected as D-NF and vice-versa, FN (false negative) number of D-NF images detected as ND-NF.[33]

Table 5: Synthetic data distribution of the three sets of synthetic datasets (DB_s) of size 500, 1600, 3196 produced by the proposed c-GAN.

Class	DB_{s1}	DB_{s2}	DB_{s3}
D-NF	240	768	1536
ND-NF	260	832	1660
Total	**500**	**1600**	**3196**

2.5. *Synthetic SEM image analysis*

Frechet Inception Distance (FID). FID measures the distance between the synthetic images and the real ones, using the the activations from the last pooling layer of the inception v3 model.[31] The FID score is calculated as follows:

$$FID = ||\mu_1 – \mu_2||^2 + Tr(C_1 + C_2 – 2(C_1C_2)^{1/2}) \quad (3)$$

where, μ_1 and μ_2 represent the feature-wise average of the real and synthetic vectors, respectively; C_1 and C_2 refer to the covariance matrices of the real and synthetic features arrays; Tr is the trace linear algebra operation. Low FID scores denote better quality of generated images, vice-versa, higher values denote lower quality of generated images.

Probability Density Function (PDF). Let I be an image with P pixels and intensity values ranged between 0 and $M-1$ (where M is the highest number of intensity levels), the histogram of I is defined as $hist(i) = p(i)$, where i denotes the intensity level, whereas, $p(i)$ is the number of pixels at the corresponding intensity level.[34] Hence, the PDF is calculated by means of the normalized histogram as follows: $PDF(i) = p(i)/P$.

t- distributed stochastic neighbor embedding (t-SNE). It is a statistical method that operates a non-linear dimensionality reduction with the aim of mapping high-dimensional data onto a low-dimensional space of two or three dimensions. Specifically, t-sNE maps each high-dimensional object onto a two- or three-dimensional point so that similar objects are projected into nearby points and dissimilar objects are projected into distant points by converting the euclidean distance between data points into joint probabilities that account similarities.[35] In this paper, the dimensionality of the features extracted by the CNN from both real and fake images was reduced

by embedding the feature data in a bidimensional space, by means of t-SNE algorithm. Data visualization in the t-SNE projected space will help to infer if features from real and fake images are projected into relatively distant or close points, in other words, if real and fake images appear distinguishable in the t-SNE projected space.

3. Results

3.1. *pre-trained CNN performance*

Table 7 reports comparative classification performance of different CNN configurations when the real D-NF/ND-NF SEM images dataset is used as train/test set. As can be noted, the lowest results were achieved with CNN_2 with an accuracy rate up to 69.62±2.94%; while, the highest performance was achieved with CNN_5 with Sp of 88.21±4.52%, Ss of 73.67±2.95 PPV of 85.59±4.84%, NPV of 78.40±1.42%, FS of 79.05±1.55% and accuracy rate up to 81.22±1.66%.

3.2. *Analysis of synthetic SEM images*

Fig. 6a and 6b show examples of D-NF and ND-NF images generated by the developed c-GAN. From visual inspection, synthetic samples look similar to images acquired from SEM analysis of electropun nanofibers produced by electrospinning experiments. In order to evaluate the quality of the synthetic images the FID and PDF were calculated. Table 6 reports a comparative analysis of FID, estimated per class, by each generative network, over different sets of synthetic data DB_{s1},DB_{s2},DB_{s3} (sized 500, 1600, 3196, respectively). As can be seen, the proposed c-GAN_3 (herein referred to as c-GAN) achieved the lowest score (i.e., average FID=85.96), using the same number of real images. Furthermore, Fig. 9a reports the comparison between the PDF of *real* D-NF SEM images and the PDF of *synthetic* D-NF SEM images; vice-versa, Fig. 9b reports PDF comparison of *real* and *synthetic* ND-NF images. As can be observed, the proposed c-GAN was able to generate D-NF/ND-NF SEM images with similar PDF as the original ones. Such result was also confirmed by the t-SNE[35] statistical method. Specifically, the dimensionality of the feature space was reduced by means of t-SNE by embedding the data in a bidimensional space. Fig. 4 shows the results of t-SNE application to the dataset of features extracted from real and synthetic images (generated by the proposed c-GAN) representing nanofibers with defects (a) and with no defects (b). Real data are represented by red points whereas synthetic data are represented by blue points. Fig. 4 shows that the scatter plots of the two classes (real and synthetic) look overlapped in the embedding space in this way endorsing that non significant differences between real and synthetic data could be detected. However, t-SNE depends on perplexity, a parameter related to the number of close neighbours of every single point.[36] Fig. 4 shows an example with perplexity set at 30. Perplexity is defined as 2 to the power of the Shannon entropy of the probability distribution P_i induced by a variance σ_i assigned to a Gaussian centered over a single high-dimensional point x_i. In order to assess whether the interpretation of the achieved results was sensitive to perplexity, the Loss and the distance between the centers of two islands (clouds of nearby points in the scatter plot) accounting for real and synthesized data were estimated for varying values of perplexity. Perplexity ranged between 5 and 50,[36] with a step of 5. The loss represents the Kullback-Leibler divergence between the joint distributions that model the input data and the embedded data. The distance between the centers of the two islands (real data vs synthetic data) was introduced to quantify how much the two islands are co-centered. The distance was on average 0.9 for DNF data and of 0.91 for NDNF data, showing overall a decreasing trend as perplexity increased, for perplexity values larger than 15 (Fig. 5). Loss also showed a decreasing trend as perplexity increased for perplexity values larger than 10, with an average value of 1.79 for DNF data and of 1.78 for NDNF data (Fig. 5).

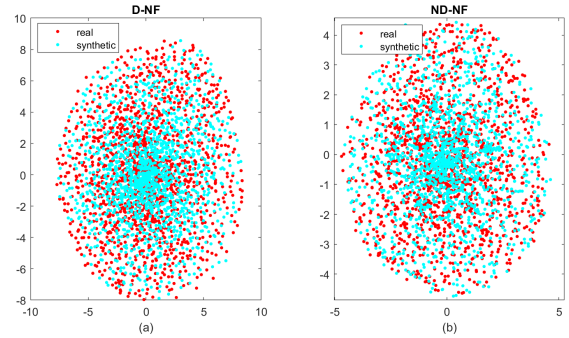


Figure 4: t-SNE visualization of the features related to real and synthetic images of NF with defects (a) and with no defects (b).

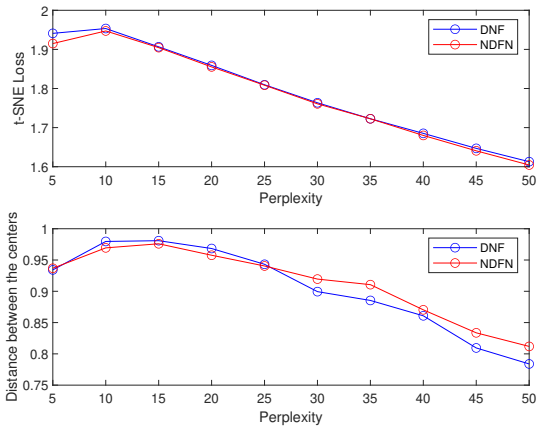


Figure 5: (Top) Trend of the t-SNE Loss (Kullback-Leibler divergence between the joint distributions that model the input data and the embedded data) for different values of perplexity. (Bottom) Trend of the distance between the centers of the two islands (real data vs synthetic data). The blue line represents the embedding of DNF data, the red line represents the embedding of NDNF data.

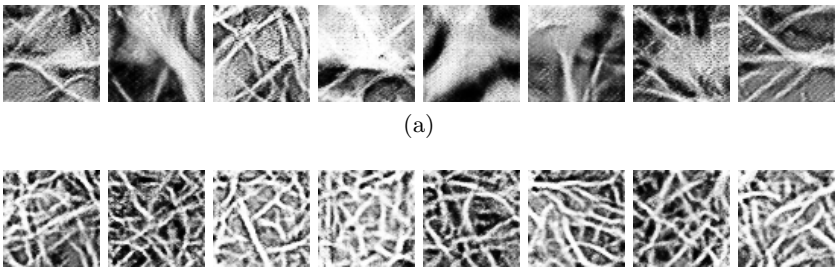
In order to further compare the real and synthetic images quantitatively, a texture analysis was proposed. To this end, a statistical method of examining texture that considers the spatial relationship of pixels known as the gray-level co-occurrence matrix (GLCM) was adopted.[37] GLCM characterizes the texture of an image by computing how often pairs of pixels with specific values and in a specific spatial relationship occur in an image. A co-occurrence matrix GLCM was calculated for every available image in the dataset of real and synthetic images. Some measures were also calculated from the co-occurrence GLCM matrices in order to quantify the texture of the images, namely: contrast (measures the local variations in the gray-level GLCM matrix); correlation (measures the joint probability occurrence of pixel pairs); energy (provides the sum of squared elements in the GLCM matrix, represents the image uniformity or the angular second moment.); homogeneity (measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal.) The texture of real and synthetic images was compared in terms of the aforementioned measures. For every measure, the values estimated from the real images were compared to those estimated from synthetic images by means of the Wilcoxon ranksum test.[38] The following p values were calculated: contrast (p=6.2*10^{-6}), correlation (p=0.13), energy (p=0.04), homogeneity (p=7.67*10^{-10}); therefore all below the threshold 0.05, except the correlation, which means that real and synthetic images exhibited no statistically significant differences in terms of contrast, energy and homogeneity of the GLCM matrices.

Table 6: Comparative analysis of FID, estimated per class, by each generative network (c-GAN$_1$, c-GAN$_2$, c-GAN$_3$).

	Class	FID-DB_{s1}	FID-DB_{s2}	FID-DB_{s3}
c-GAN$_1$	D-NF	291,36	266,83	252,49
	ND-NF	300,7	305,9	303,72
	average	**296,03**	**286,36**	**278,10**
c-GAN$_2$	D-NF	184,92	146	121,59
	ND-NF	191	171,18	155
	average	**188,1**	**158,72**	**138,41**
c-GAN$_3$	D-NF	203,01	117,35	83
	ND-NF	203,35	129,31	89
	average	**203,18**	**123,33**	**85,96**

3.3. *transfer-learned CNN performance*

The CNN_5 configuration, i.e., the trained CNN_5 achieving the maximum accuracy across the different k-fold runs, was used as *pre-trained CNN* to perform transfer learning. Note that the *transfer-learned CNN* was trained only using synthetic images generated by the proposed *c-GAN* and tested on real images. Furthermore, the performance metrics were estimated with various number of training samples. Indeed, the *transfer-learned CNN* was trained over the three different sets of synthetic images (500, 1600, 3196) generated by the proposed c-GAN (Section 2.2.2). Table 8, reports comparative performance, including computational cost. As can be seen, higher classification performance was achieved by increasing the number of the training samples to the detriment of processing time. Indeed, accuracy was of 95.31%



(a)

(b)

Figure 6: (a) Examples of *synthetic* D-NF SEM images. (b) Examples of *synthetic* ND-NF SEM images

with a computational cost of 124 s. More specifically, Fig. 7 reports the confusion matrix with the number of FP, FN, TP, TN. Hence, the transfer learning based configuration allowed to improve the classification performance of about 10% when compared with the *pre-trained CNN*, Table 9. This results was also confirmed by the analysis of the Area Under the Receiver Operating Characteristic (ROC) curve (AUROC), as reported in Fig. 8, with AUROC of 99.04% and 91.04% for the *transfer-learned CNN* and *pre-trained CNN*, respectively. In addition, classification metrics have been estimated also shuffling real and synthetic images since the beginning and compared with the *transfer-learned CNN*. The mixed data were used to train and test the customized CNN, reporting accuracy rate up to 84.59%, as reported in Table 10. This result confirms the usefulness of the proposed *transfer learning-oriented* approach that allowed to achieve higher performance (i.e., 95.31%).

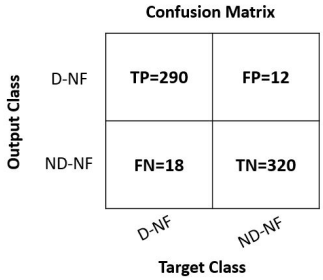


Figure 7: Confusion matrix of the proposed *transfer-learned CNN*.

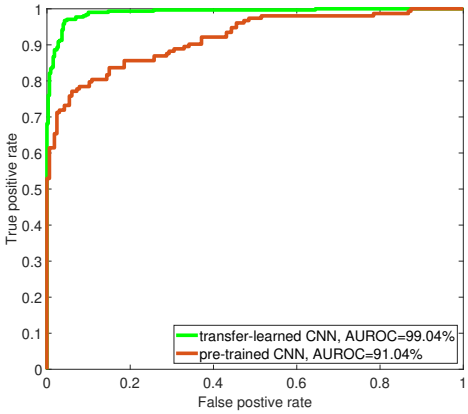
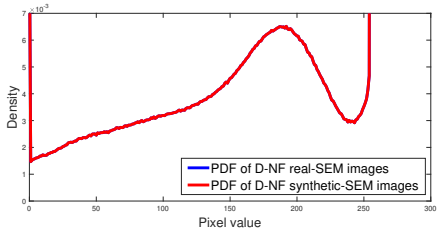


Figure 8: ROC curves and AUC values of the *pre-trained CNN* and the proposed *transfer-learned CNN*.

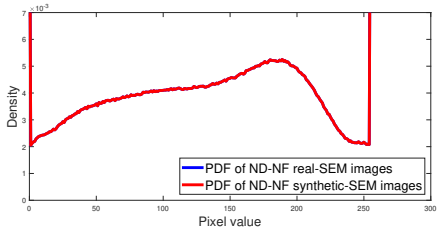
3.4. *Comparison with the state-of-the-art*

To the best of our knowledge, there are few works related to AI for processing SEM images of defective and non-defective NFs produced by electrospinning process (Table 11). Some authors[12,39] developed AI-based systems for detecting and localizing defects; others,[13,14] focused on the classification of D-NF/ND-NF SEM images, while, there is a limited number of works that used augmentation strategies for nanomaterial applications. Specifically, in[15] in order to increase the cardinality of the dataset, features extracted from patches (sized 64 x 64) by a unsupervised processor (i.e., AE) were corrupted by white Gaussian noise and used as input to a supervised processor (i.e., MLP) achieving an accuracy rate up to 92.5%. In[16] common augmentation strategies (noise, translations and rotations) were applied, reporting accuracy rate up to 93.85%±1.65%. To the best of our knowledge, this is the very first work that

proposes a c-GAN for synthetic SEM image generation. Newly generated (synthetic) data were then used in downstream tasks to reduce the need to perform expensive experiments to obtain sufficient data. This was supported by the use of a *transfer-learned CNN*, trained on synthetic images and validated on real ones, achieving an accuracy rate of up to 95.31% and AUROC up to 99.04%, thus outperforming discrimination scores of other models that do not rely on generative networks as reported in Table 11.



(a)



(b)

Figure 9: (a) Comparison of the probability density function of D-NF *real*-SEM images and D-NF *synthetic*-SEM images. (b) Comparison of the probability density function of ND-NF *real*-SEM images and ND-NF *synthetic*-SEM images.

4. Discussion

The original contribution of the present work lies in proposing a conditional generative adversarial network to model electrospun nanofiber data. Newly generated (synthetic) data were then used in downstream tasks to reduce the need to perform expensive experiments to obtain sufficient data. To this end, a *transfer-learned CNN* was proposed, achieving an accuracy rate of up to 95.31%, thus outperforming discrimination scores of other models that do not rely on generative networks. It is to be noted that a *transfer learning-oriented* approach is presented, indeed, the knowledge learned to distinguish real D-NF/ND-NF SEM images forms the basis of an improved model that aims to classify D-NF/ND-NF images merely relying on totally new synthetic samples. Such strategy allowed not only to improve the classification performance but also to demonstrate that synthetic images, generated by the proposed c-GAN, had similar properties of real samples. This was also confirmed by the comparative analysis of the probability density functions (shown in Fig. 9a and 9b) and FID score (Table 6). However, the proposed approach has some limitations. First, the proposed c-GAN was not able to reproduce the original full-size SEM image sized 128 x 128. This was possibly due to the limited size D-NF and ND-NF samples used to train the generative model. Second, the original image sized 128 x 128 was divided into four sub-images, each manually annotated as D-NF and DN-NF. As a consequence, the procedure was initially dependent on an expert operator. In addition, the quality of anomalous and normal images generated by the c-GAN have been estimated by means of the FID score. In particular, FID values of 83 and 89 were achieved for D-NF and ND-NF class. Indeed, from a visual inspection analysis, it was noted that some synthetic SEM images generated as ND-NF included small defects (and vice-versa) instead. This was possibly due to the fact that the system was not able to recognize and reproduce such defects and, as consequence, some images labelled as ND-NF were generated with small anomalies. In addition, some electrospinning experiments produced nanofibers with large diameters which could mimic the presence of defective, confusing the network, as a consequence, some synthetic images were labeled as D-NF.

5. Conclusion

In this study, a generative adversarial-based approach is proposed to generate synthetic SEM images of nanofibers produced by electrospinning process, with the ultimate aim of avoiding the costly laboratory experiments. In particular, a conditional-GAN was developed to produce SEM images of defective and non-defective nanofibers, allowing the reconstruction of high fidelity SEM images. To the best of our knowledge, this is the first work that employs adversarial networks for the generation of realistic-looking SEM images. Furthermore, transfer learning strategy was also applied by means of a *pre-trained CNN* in order to transfer the extracted "knowledge" (i.e., weights) of the CNN processing modules, from *real*-SEM images, to the new network, here referred

Table 7: Setup of different CNN configurations and comparative classification performance

Model	$conv_1$+**ReLU**	$mpool_1$	$conv_2$+**ReLU**	$mpool_2$	**HL1**	**HL2**	**Specificity** [%]	**Sensitivity** [%]	**PPV** [%]	**NPV** [%]	**F-Score** [%]	**Accuracy** [%]
CNN_1	Filters=8@4x4, s=2	2x2, s=2	-	-	100	-	79.35±6.83	61.50±9.49	73.99±4.51	69.37±3.72	66.57±5.44	70.78±2.81
CNN_2	Filters=8@4x4, s=2	2x2, s=2	-	-	100	50	75.22±13.82	63.60±12.98	72.33±7.69	69.88±4.50	66.30±5.30	69.62±2.94
CNN_3	Filters=8@4x4, s=2	2x2, s=2	Filters=16@4x4, s=2	2x2, s=2	30	-	87.77±2.39	67.94±3.64	85.06±2.85	75.21±2.13	75.88±2.76	79.28±2.12
CNN_4	Filters=8@4x4, s=2	2x2, s=2	Filters=16@4x4, s=2	2x2, s=2	30	10	87.24±2.64	70.09±3.58	83.67±2.67	75.98±2.00	76.21±2.13	79.00±1.61
CNN_5	**Filters=64@4x4, s=2**	**2x2, s=2**	**Filters=96@4x4, s=2**	**2x2, s=2**	**1000**	-	**88.21±4.52**	**73.67±2.95**	**85.59±4.84**	**78.40±1.42**	**79.05±1.55**	**81.22±1.66**
CNN_6	Filters=64@4x4, s=2	2x2, s=2	Filters=96@4x4, s=2	2x2, s=2	1000	50	87.54±5.58	73.15±7.52	84.98±4.36	78.24±4.08	78.25±3.44	80.62±2.28

Table 8: Comparative analysis of the proposed *transfer-learned CNN* in terms of performance and computational cost with various number of training samples. Results in the table refer to the test set (real images). Comparative results of the proposed *pre-trained* CNN and *transfer-learned* CNN

Training samples	**Specificity** [%]	**Sensitivity** [%]	**PPV** [%]	**NPV** [%]	**F-Score** [%]	**Accuracy** [%]	**Time Cost** [s]
DBs1 (500 synthetic images)	96.08%	89.61%	95.50%	90.88%	92.46%	92.97%	22.92
DBs2 (1600 synthetic images)	95.78%	92.53%	95.32%	93.26%	93.90%	94.22%	58.18
DBs3 (3196 synthetic images)	96.39%	94.16%	96.03%	94.67%	95.08%	95.31%	124.11

Table 9: Comparative results of the proposed *pre-trained* CNN and *transfer-learned* CNN

Model	**Specificity** [%]	**Sensitivity** [%]	**PPV** [%]	**NPV** [%]	**F-Score** [%]	**Accuracy** [%]
pre-trained CNN	97.59%	70.78%	96.46%	78.26%	81.65%	84.69%
transfer-learned CNN	96.39%	94.16%	96.03%	94.67%	95.08%	95.31%

to as *transfer-learned CNN*. This procedure will reduce the computational effort of the successive training step, as most of the features representing the problem have already been extracted. Overall, the achieved promising results of the proposed generative system would allow, in principle, for applications in the industrial deployment in nanotechnology scenario. Indeed, the possibility to generate synthetic D-NF and ND-NF SEM would augment the dataset needed to train an automatic anomaly detection system, avoiding to run additional costly electrospinning experiments and improving the monitoring of nanomaterials in the production chain. In the future, we intend to carry out new electrospinning experiments using different polymeric solutions in order to further increase the cardinality of the original dataset. Furthermore, modern neural network structures (e.g., transformers) as well as more robust generative models and advanced supervised machine learning/classification algorithms such as neural dynamic classification algorithm,[41,42] dynamic ensemble learning algorithm,[43] and finite element machine for fast learning,[44] will be explored in an attempt to improve the detection performance.

6. Acknowledgements

The authors would like to thank the researchers of the *Materials for Environmental and Energy Sustainability Laboratory* at the University Mediterranea of Reggio Calabria (Italy) for the use of the electrospinning apparatus to produce the SEM image dataset employed in this work. This paper was supported by the Programma Operativo Nazionale (PON) "Ricerca e Innovazione" 2014-2020 CCI2014IT16M2OP005, code: I05.

Bibliography

1. W. E. Teo and S. Ramakrishna, A review on electrospinning design and nanofibre assemblies, *Nanotechnology* **17**(14) (2006) p. R89.
2. J. H. Wendorff, S. Agarwal and A. Greiner, *Electrospinning: materials, processing, and applications* (John Wiley & Sons, 2012).
3. C. Ieracitano, N. Mammone, M. Versaci, G. Varone, A.-R. Ali, A. Armentano, G. Calabrese, A. Ferrarelli, L. Turano, C. Tebala *et al.*, A fuzzy-enhanced deep learning approach for early detection of covid-19

Table 10: Comparative results of the proposed *customized* CNN and *transfer-learned* CNN.

Model	**Specificity** [%]	**Sensitivity** [%]	**PPV** [%]	**NPV** [%]	**F-Score** [%]	**Accuracy** [%]
customized CNN	92.19%	81.76%	90.61%	84.57%	85.96%	87.19%
transfer-learned CNN	96.39%	94.16%	96.03%	94.67%	95.08%	95.31%

Table 11: Comparative analysis with existing approaches.

Ref.	**Dataset**	**Methodology**	**Purpose**	**Results**
Carrera et al.[40]	20 SEM images (40 D-NF; 5 N-D-NF) 1024 × 700	Multiscale group-sparse coding and indicator	Anomaly Detection/Localization	AUROC: 90.6%
Carrera et al.[9]	45 SEM images (40 D-NF; 5 N-D-NF) 1024 × 700	Sparse representation + patch-wise manner	Anomaly Detection/Localization	AUROC: 93%
Napoletano et al.[12]	45 SEM images (40 D-NF; 5 N-D-NF) 1024 × 700	CNN-based self similarity	Anomaly Detection/Localization	AUROC: 97%
Bergmann et al.[39]	150 SEM images (100 D-NF; 50 N-D-NF) 512 × 512	Structural similarity-based AE	Anomaly Detection/Localization	AUROC: 96%
Ieracitano et al.[13]	160 SEM images (85 D-NF; 75 N-D-NF) (size 128x128)	Raw data + deep CNN	D-NF/N-D-NF Classification	Acc: 80%
Ieracitano et al.[14]	160 SEM images (85 D-NF; 75 N-D-NF) (size 128x128)	Sobel filtering + deep CNN (SoCNNet)	D-NF/N-D-NF Classification	Acc: 80.27%
Ieracitano et al.[15]	640 SEM images (320 D-NF; 320 N-D-NF) (size 64x64)	Hybrid unsupervised/supervised Machine Learning (AE+MLP)	D-NF/N-D-NF Classification	Acc: 92.5%
Ieracitano et al.[16]	160 SEM images (85 D-NF; 75 N-D-NF) (size 128x128)	Standard data augmentation + deep CNN	D-NF/N-D-NF Classification	Acc: 93.85%
Proposed methodology	**~3200 SEM images (size 64 x 64)**	**c-GAN + transfer learning-oriented approach + deep CNN**	**synthetic SEM image generation and D-NF/N-D-NF Classification**	**Acc: 95.31%, AUROC: 99.04%**

pneumonia from portable chest x-ray images, *Neurocomputing* (2022).
4. G. Alexandridis, J. Aliprantis, K. Michalakis, K. Korovesis, P. Tsantilas and G. Caridakis, A knowledge-based deep learning architecture for aspect-based sentiment analysis, *International journal of neural systems* **31**(10) (2021) p. 2150046.
5. D. S. Jodas, T. Yojo, S. Brazolin, G. D. N. Velasco and J. P. Papa, Detection of trees on street-view images using a convolutional neural network, *International journal of neural systems* **32**(01) (2022) p. 2150042.
6. W. Liu, Y. Wu, Y. Hong, Z. Zhang, Y. Yue and J. Zhang, Applications of machine learning in computational nanotechnology, *Nanotechnology* **33**(16) (2022) p. 162501.
7. S. Kulkarni, S. Bhat and C. A. Moritz, Architecting for artificial intelligence with emerging nanotechnology, *ACM Journal on Emerging Technologies in Computing Systems (JETC)* **17**(3) (2021) 1–33.
8. G. Boracchi, D. Carrera and B. Wohlberg, Novelty detection in images by sparse representations, *2014 IEEE Symposium on Intelligent Embedded Systems (IES)*, IEEE2014, pp. 47–54.
9. D. Carrera, F. Manganini, G. Boracchi and E. Lanzarone, Defect detection in SEM images of nanofibrous materials, *IEEE Transactions on Industrial Informatics* **13**(2) (2016) 551–561.
10. A. Adler, M. Elad, Y. Hel-Or and E. Rivlin, Sparse coding with anomaly detection, *Journal of Signal Processing Systems* **79**(2) (2015) 179–188.
11. J. Zujovic, T. N. Pappas and D. L. Neuhoff, Structural texture similarity metrics for image analysis and retrieval, *IEEE Transactions on Image Processing* **22**(7) (2013) 2545–2558.
12. P. Napoletano, F. Piccoli and R. Schettini, Anomaly detection in nanofibrous materials by CNN-based self-similarity, *Sensors* **18**(1) (2018) p. 209.
13. C. Ieracitano, F. Pantó, N. Mammone, A. Paviglianiti, P. Frontera and F. C. Morabito, Toward an automatic classification of SEM images of nanomaterials via a deep learning approach, *Neural Approaches to Dynamics of Signal Exchanges*, (Springer, 2020), pp. 61–72.
14. C. Ieracitano, A. Paviglianiti, N. Mammone, M. Versaci, E. Pasero and F. C. Morabito, Socnnet: An optimized sobel filter based convolutional neural network for SEM images classification of nanomaterials, *Progresses in Artificial Intelligence and Neural Systems*, (Springer, 2021), pp. 103–113.
15. C. Ieracitano, F. C. Morabito, A. Hussain and N. Mammone, A hybrid-domain deep learning-based bci for discriminating hand motion planning from eeg sources, *International journal of neural systems* **31**(09) (2021) p. 2150038.
16. C. Ieracitano, N. Mammone, A. Paviglianiti and F. C. Morabito, Toward an augmented and explainable machine learning approach for classification of defective nanomaterial patches, *International Conference on Engineering Applications of Neural Networks*, Springer2021, pp. 244–255.
17. K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng and F.-Y. Wang, Generative adversarial networks: introduction and outlook, *IEEE/CAA Journal of Auto-*

matica Sinica **4**(4) (2017) 588–598.
18. L. Torrey and J. Shavlik, Transfer learning, *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, (IGI global, 2010), pp. 242–264.
19. S. Coppola, V. Vespini, G. Nasti, O. Gennari, S. Grilli, M. Ventre, M. Iannone, P. A. Netti and P. Ferraro, Tethered pyro-electrohydrodynamic spinning for patterning well-ordered structures at micro- and nanoscale, *Chemistry of Materials* **26**(11) (2014) 3357–3360.
20. L. Zhang and T. J. Webster, Nanotechnology and nanomaterials: promises for improved tissue regeneration, *Nano today* **4**(1) (2009) 66–80.
21. R. S. Bhattarai, R. D. Bachu, S. H. Boddu and S. Bhaduri, Biomedical applications of electrospun nanofibers: Drug and nanoparticle delivery, *Pharmaceutics* **11**(1) (2019) p. 5.
22. M. Shi, L. Zhao, X. Song, J. Liu, P. Zhang and L. Gao, Highly conductive mo2c nanofibers encapsulated in ultrathin mno2 nanosheets as a self-supported electrode for high-performance capacitive energy storage, *ACS applied materials & interfaces* **8**(47) (2016) 32460–32467.
23. Z.-M. Huang, Y.-Z. Zhang, M. Kotaki and S. Ramakrishna, A review on polymer nanofibers by electrospinning and their applications in nanocomposites, *Composites science and technology* **63**(15) (2003) 2223–2253.
24. M. M. Demir, I. Yilgor, E. Yilgor and B. Erman, Electrospinning of polyurethane fibers, *Polymer* **43**(11) (2002) 3303–3309.
25. N. Bhardwaj and S. C. Kundu, Electrospinning: a fascinating fiber fabrication technique, *Biotechnology advances* **28**(3) (2010) 325–347.
26. Z.-M. Huang, Y. Zhang, S. Ramakrishna and C. Lim, Electrospinning and mechanical characterization of gelatin nanofibers, *Polymer* **45**(15) (2004) 5361–5368.
27. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, Generative adversarial nets, *Advances in neural information processing systems* **27** (2014).
28. M. Mirza and S. Osindero, Conditional generative adversarial nets, *arXiv preprint arXiv:1411.1784* (2014).
29. S. Albawi, T. A. Mohammed and S. Al-Zawi, Understanding of a convolutional neural network, *2017 International Conference on Engineering and Technology (ICET)*, Ieee2017, pp. 1–6.
30. A. Radford, L. Metz and S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, *arXiv preprint arXiv:1511.06434* (2015).
31. M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler and S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, *Advances in neural information processing systems* **30** (2017).
32. C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang and C. Liu, A survey on deep transfer learning, *International conference on artificial neural networks*, Springer2018, pp. 270–279.
33. W. Zhu, N. Zeng, N. Wang *et al.*, Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas implementations, *NESUG proceedings: health care and life sciences, Baltimore, Maryland* **19** (2010) p. 67.
34. A. K. Bhandari, K. Srinivas and A. Kumar, Optimized histogram computation model using cuckoo search for color image contrast distortion, *Digital Signal Processing* **118** (2021) p. 103203.
35. G. E. Hinton and S. Roweis, Stochastic neighbor embedding, *Advances in neural information processing systems* **15** (2002).
36. L. Van der Maaten and G. Hinton, Visualizing data using t-sne., *Journal of machine learning research* **9**(11) (2008).
37. R. M. Haralick, K. Shanmugam and I. H. Dinstein, Textural features for image classification, *IEEE Transactions on systems, man, and cybernetics* (6) (1973) 610–621.
38. J. D. Gibbons and S. Chakraborti, *Nonparametric statistical inference* (CRC press, 2014).
39. P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger and C. Steger, Improving unsupervised defect segmentation by applying structural similarity to autoencoders, *arXiv preprint arXiv:1807.02011* (2018).
40. D. Carrera, G. Boracchi, A. Foi and B. Wohlberg, Scale-invariant anomaly detection with multiscale group-sparse models, *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE2016, pp. 3892–3896.
41. M. Ahmadlou and H. Adeli, Enhanced probabilistic neural network with local decision circles: A robust classifier, *Integrated Computer-Aided Engineering* **17**(3) (2010) 197–210.
42. M. H. Rafiei and H. Adeli, A new neural dynamic classification algorithm, *IEEE transactions on neural networks and learning systems* **28**(12) (2017) 3074–3083.
43. K. M. Alam, N. Siddique, H. Adeli *et al.*, A dynamic ensemble learning algorithm for neural networks, *Neural Computing and Applications* **32**(12) (2020) 8675–8690.
44. D. R. Pereira, M. A. Piteri, A. N. Souza, J. P. Papa and H. Adeli, Fema: A finite element machine for fast learning, *Neural Computing and Applications* **32**(10) (2020) 6393–6404.