Project Acronym:	STAR
Grant Agreement number:	956573 (H2020-ICT-2020-1 – Research and Innovation Action)
Project Full Title:	Safe and Trusted Human Centric Artificial Intelligence in Future Manufacturing Lines
Project Coordinator:	INTRASOFT International

Funded by the Horizon 2020 Framework Programme of the European Union

DELIVERABLE

D4.3 – Simulated Reality for Human Robot Collaboration

Dissemination level	PU -Public
Type of Document	Demonstrator
Contractual date of delivery	31/03/2023
Deliverable Leader	UPRC
Status - version, date	Final – v1.0, 12/04/2023
WP / Task responsible	WP4
Keywords:	Simulated Reality

Executive Summary

This deliverable presents the progress and outcomes of T4.2 carried out throughout the months M03-M27 of the project. The goal of the task was to develop tools that will enable AI models to operate in real-life environments, which are dynamic and unpredictable and where also limited data is available.

To address these goals, research of the current state-of-the-art was conducted in M3-M6 and continued through the rest of the task's duration as needed. Months M06-M15 were devoted mainly to addressing the issue of limited data, while during the period M15-27 the focus shifted to addressing the issue of resilience to unpredictable inputs.

The structure of the deliverable corresponds to the above two strands of research and development, which are also reflected in various submitted publications to scientific journals and conferences (e.g., [REF-01], [REF-02], [REF-03]). Implementations thereof can be found at https://github.com/tspyrosk/oversampling-defect-recognition and https://github.com/tspyrosk/osr-data-augmentation.

Deliverable Leader:	UPRC
Contributors:	UPRC, QLE
Reviewers:	SUPSI, UNP
Approved by:	Charalampos Ipektsidis (INTRA)

Document History			
Version	**Date**	**Contributor(s)**	**Description**
0.1	02/03/2023	UPRC	TOC
0.2	23/03/2023	UPRC, QLE	First Draft
0.3	28/03/2023	UPRC	Minor Corrections
0.4	06/04/2023	UNP, SUPSI	Review
0.5	10/04/2023	UPRC, QLE	Address Review Comments
1.0	12/04/2023	INTRA	QA and creation of the final submitted version

Table of Contents

Table of Figures

List of Tables

Definitions, Acronyms and Abbreviations

Acronym/ Abbreviation	Title
AI	Artificial Intelligence
AUROC	Area Under the Receiver Operating Characteristic
CNN	Convolutional Neural Network
CPU	Central Process Unit
CVAE	Convolutional Variational Autoencoder
DCGAN	Deep Convolutional Generative Adversarial Network
DFKDE	Deep Feature Kernel Density Estimation
DFM	Deep Feature Modelling
DL	Deep Learning
DNN	Deep Neural Network
EVT	Extreme Value Theory
GAN	Generative Adversarial Network
MLP	Multi-Layer Perceptron
OSR	Open Set Recognition
OSRCI	Open Set Recognition with Counterfactual Images
PCA	Principal Component Analysis
PI-SVM	Probability of Inclusion - Support Vector Machines
RAM	Random Access Memory
RBF	Radial Basis Function
SeFa	Semantic Factorization
VGG	Visual Geometry Group
W-SVM	Weibull - Support Vector Machines
WP	Work Package

1 Introduction

In the context of STAR, we focused on the PCL defect detection through human robot collaboration use case in which we identified the following three issues that could be addressed through the simulation of image inputs:

1. Insufficiency of training data, especially regarding rarely-occurring production defects leading to datasets with class imbalance (many in-quality products, just a few defective products), which makes it harder to train AI models.
2. High visual similarity between good and defective products making them hard to distinguish automatically.
3. Occurrence of unanticipated defects and other errors during the continuous operation phase, which can lead to wrong AI decisions since they lie outside of the algorithm's training domain.

Our selection and evaluation of algorithms was tailored to a human robot collaboration scenario, where humans will be responsible mainly for checking suspected defects only (and helping the AI learn better through active learning). A large part of our work for recognizing novel defects can also be repurposed for other scenarios where AI decisions are based on visual inputs, such as in robotic perception and manipulation.

The deliverable consists of two chapters, the first describing our approach to addressing the class imbalance problem (issue 1.) through confidence-aware GAN-based data augmentation and the second on how highly generalizable GAN architectures can be leveraged to generate out-of-distribution images (issue 2.) and make visual classifiers more robust to novel inputs (issue 3.).

2 Simulation of Image Inputs for Imbalanced Datasets

This part of our work is reflected in more detail in [REF-01] and its implementation can be found in https://github.com/tspyrosk/oversampling-defect-recognition.

2.1 Background

Automated visual quality inspection datasets suffer very often from class imbalances. This is due to the fact that modern manufacturing processes nowadays produce very few defects [REF-04]. Collecting sufficient examples of these defects to train a modern AI algorithm such as a Convolutional Neural Network (CNN), has proven exceptionally labor-intensive and costly, given that often thousands to tens of thousands of image samples are required. Nevertheless, tackling the problem of automating visual quality inspection in a way that relies on the synergy between humans and AI has great value for reducing inspection costs and freeing up human resources for performing more demanding and less repetitive [REF-05].

A first step in this direction is addressing the class imbalance issue so that the AI algorithms can provide a satisfactory baseline performance upon which techniques for synergistic learning between human operators and AI can improve. It is important to mention here that methods based on deep learning such as Convolutional Neural Networks (CNNs) are of preference in many scenarios, where flexibility and adaptability are key requirements, as these algorithms have no need for pre-extracted features (which are usually tailored to one product and hard to adapt to another) and are to a large extent uninfluenced by differences in translation and scale [REF-06]. As mentioned above the price paid for this flexibility is the large amount of required training samples and the resulting sensitivity to class imbalance.

An initial approach towards a solution is to reuse the knowledge representation already learnt by large pre-trained networks, e.g., Imagenet, which are usually trained on diverse and very large datasets. The pre-trained network is then fine-tuned to the more specific problem at hand, namely defect detection, by feeding the network with a dataset sampled by the current scenario (e.g., images of defective products). We observed however that such an approach, though sometimes beneficial, does not always work because Imagenet categories, which differ substantially from each other, do not resemble the case of defective vs non-defective products, which are instead very similar visually, and therefore harder to differentiate.

A more direct approach would be to oversample the minority classes with artificially generated images, an approach usually referred to as data augmentation and largely adopted in the scientific literature (e.g., [REF-07], [REF-08]). In its most basic form, new images are generated through the application of graphical transformation, such as scaling, illumination, shearing, rotation, or translation. However, if higher level features are needed to separate the classes, these transformations are usually insufficient [REF-09]. Taking a step up in terms of complexity, one can use generative algorithms based on deep learning such as Convolutional Variational Autoencoders (CVAEs) and Generative Adversarial Networks (GANs) to generate images from the same distributions as the minority classes. CVAEs have been used successfully on metal surface defects [REF-10], while GANs have proven effective in a variety of applications featuring class imbalances such as person reidentification [REF-11]. Neural Style Transfer, a deep learning approach that uses a fusion network to combine a "style" image with a "content" image has been used to fuse segmented defect patches onto varied

locations of a product image [REF-12]. It should be noted that all these methods are considered data-hungry and their suitability depends on the degree of class imbalance in the dataset.

As part of the work carried out in T4.2, different approaches have been tested and compared. Firstly, pre-trained classifiers were enhanced through the use of oversampling methods such as SMOTE [REF-13], Borderline-SMOTE [REF-14] and ADASYN [REF-15], which augment the data after the transfer-learned features are extracted. Despite their satisfactory results, custom shallow CNNs that are trained exclusively on the products dataset (no transfer learning) can recall defects more successfully. Unfortunately, the "cheaper" feature vector augmentation methods were not easily applicable on shallow CNNs that are trained end-to-end due to the high dimensionality of their feature space (deeper networks have the ability to reduce dimensionality after many consecutive convolutional layers). Thus, the remaining option was to employ a deep-learning-based image generation method such as the one presented. As both CVAEs and GANs were computationally expensive, we utilized the small-sample GAN adaptation method presented in [REF-16] to produce minority class (defect) images on the fly. Also borrowing from the Borderline SMOTE method, the simulation of defects was enhanced to synthesize mostly low-confidence images that help the underlying classifier delineate better boundaries between defects and non-defects.

2.2 Methods

As described in the previous section, we aimed at applying oversampling on the level of raw images, so that we can provide to a custom shallow CNN the same improvement as vector-based oversampling does to a transfer-learning network. This would allow us to obtain performance gains both from the end-to-end training of the shallow CNN and from mitigating the class imbalance of the PCL shaver dataset. To provide some more specific context, the PCL shaver dataset consists of three classes:

- Good Images with 2684 instances
- Double Print Images with 244 instances
- Interrupted Images with 598 instances

Below we can see a diagram of the proposed approach for tackling the PCL shavers dataset and the various steps it entails.

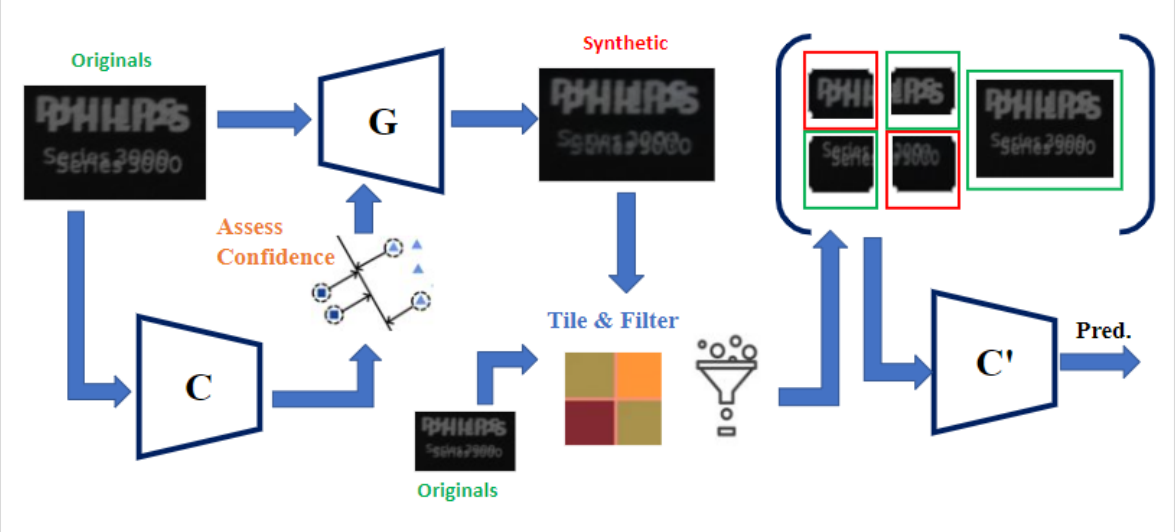


Figure 1: Image-based and confidence-aware simulation of image inputs for the PCL shaver dataset.

C is the CNN classifier trained on the original dataset without augmentation, G is the BigGAN based generator used to generate synthetic images and C' is the final classifier after training on the augmented dataset.

Firstly, an initial shallow CNN classifier C is trained on the imbalanced dataset for an initial number of epochs. The aim of this training is to provide information on which images the classifier C can classify with high confidence and which images cannot be clearly distinguished as defects or non-defects. To measure this confidence the weights of the trained classifier are used, together with the layer outputs from each specific image. All this information feeds into a function introduced in [REF-17] and shown in the figure under the "Assess Confidence" step, that tries to approximate the projected distance of a sample from the Neural Network's classification boundary. This is analogous to the distance from boundary concept in Support Vector Machines. However, DNNs have many nonlinearities and operate on very high-dimensional spaces, so in this case an approximation is needed. We consider this calculated distance a measure of the confidence of the classifier in its classification of a sample.

The next step is the generation process. This process is based on adapting BigGAN to work on small sample datasets (as presented in [REF-16]) and essentially takes the low-confidence images from the confidence assessment step and produces small variations of them, thus slightly extending the input distribution. The ultimate aim of this process is not only to balance the classes but to do so in the most informative way, by generating images close to the classifier's knowledge boundaries. The synthetic images also go through a post-processing process, which fuses them with real images through tiling. The need for this process is twofold, firstly to shift some weights away from the generation process by reducing the number of purely synthetic images needed, which though adapted is still computationally heavy; secondly, to increase the fidelity of the generated images, in case some of them have not been synthesized well-enough by the GAN. As a final post-processing step, generated images are ranked using an image similarity metric (such as SSID, MSE) and all except top-scoring images are filtered out to ensure that they belong to the correct class distribution.

Finally, an augmented dataset is constructed in which the simulated images are added to the minority classes so that the instance number is balanced across all classes. A new classifier C' of the same architecture as C is now trained from scratch on the augmented dataset and used for obtaining predictions for the test set. Figure 2 shows the architecture details for C and C'.

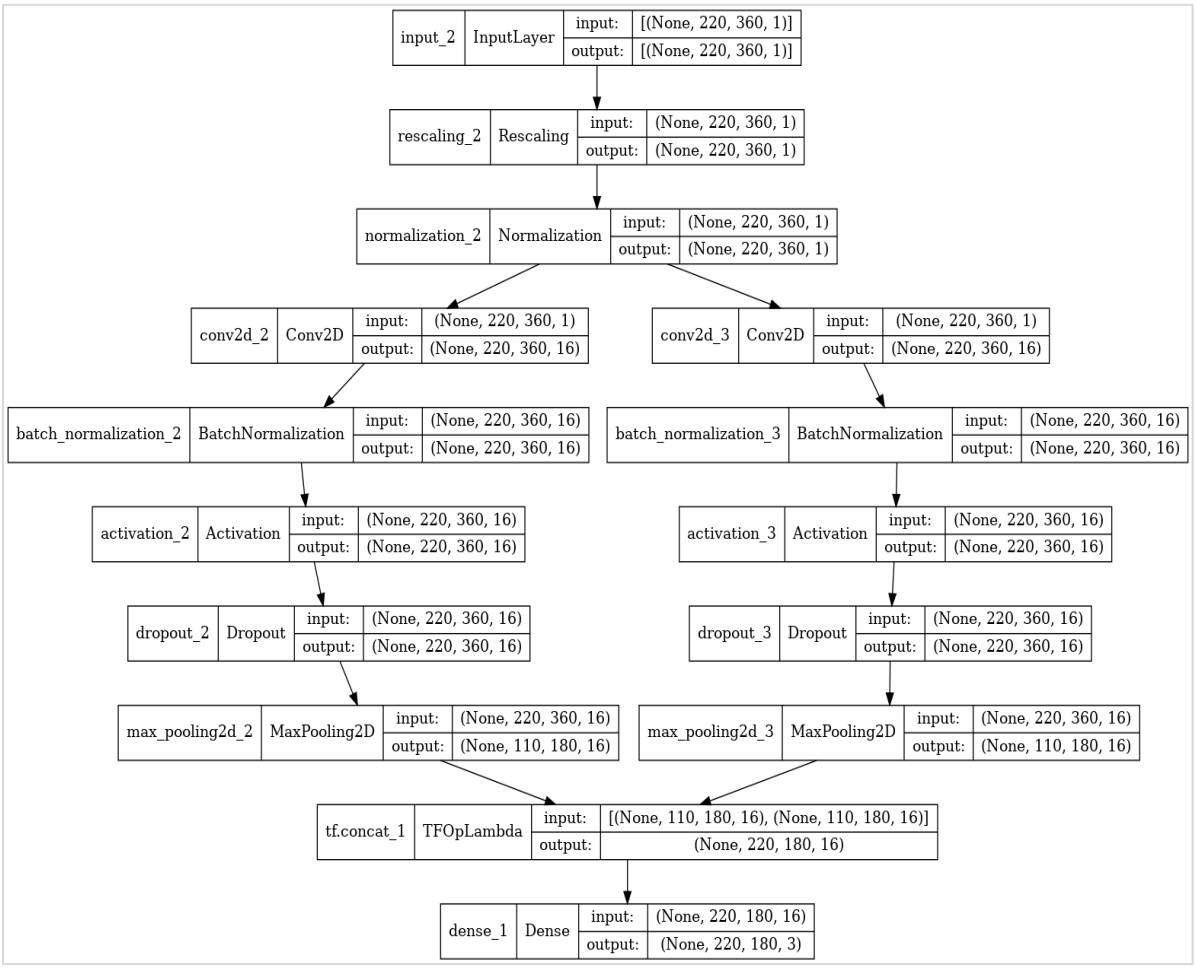


Figure 2. Custom CNN architecture of C and C'.

2.3 Evaluation

While choosing methods to apply to the problem at hand and ways to evaluate we determined three lines of approaching the problem as most critical:

1. Direct data augmentation using the highest fidelity GANs available in the Machine Learning state-of-the-art. The purpose of this approach is simple, namely, to generate as many plausible images as possible given an initial class distribution. To this end, we used Stylegan v3, an updated version of the Stylegan model [REF-18].
2. Transfer learning combined with feature vector-based oversampling. A sensible approach trying to make use of the various pre-trained classifiers from the Imagenet challenge that are available in various libraries such as tensorflow and scikit-learn.
3. Approaches that perform oversampling on the level of raw images such as DeepSMOTE [REF-19] which used Autoencoders to produce synthetic data from learning linear

interpolation between inputs in a SMOTE-like fashion. The above outlined method also falls into this category.

To evaluate the performance and compare the different methods we focused on measuring the classifier's ability to recall defects through the binary recall metric. Although our scenario is one of multi-class classification, we considered confusion between different defect classes of less importance, especially since a few of the defect instances cannot be strictly categorized (i.e. they are double printed with interruptions). Furthermore, this metric suits our use case since we want to make sure that the AI marks as many defects as possible to be later examined by humans. Mistakenly marking defects as good means that they will go through the inspection process unchecked. Traditional accuracy metrics may hide these defects as the scores are usually skewed by the majority (non-defect) classes. However, as a secondary concern, the number of good items marked as defects should also remain under control since then the human operators will be overburdened and the purpose of automation will be defeated. For this reason, we also monitor the AU-ROC as a secondary metric, which although biased towards the majority class, should alert us via a low score if many good products are marked as defects. We are of course willing to sacrifice the AU-ROC score in favor of binary recall, but we monitor it to ensure this "sacrifice" will be limited. For example, an algorithm marking everything as a defect will have perfect binary recall but very low AU-ROC and we want to avoid such an extreme case.

To conduct the experiments, 30 different runs were performed in six sets of 5-fold cross-validation runs, so that at least 20% of the instances end up in the test set of every run - this is particularly important for obtaining stable scores over a satisfactory number of minority class instances. To ensure all methods ran efficiently a NVidia K80 GPU was used for both training and the generation of synthetic data.

Table 1 shows promising results for our developed method. There are also several interesting observations regarding the experimental comparison of the DL/oversampling methods:

- Transfer Learning performance can be substantially enhanced through the use of vector-based oversampling (Borderline SMOTE, ADASYN);
- Despite the use of oversampling, custom shallow CNNs trained end-to-end on the raw inputs still manage to recognize a larger percentage of defects;
- Custom CNN's can be further enhanced by loss weighting (giving higher weight to the loss incurred by minority class samples during training);
- Confidence-aware image-level oversampling (our approach) can also provide custom CNNs with a small but substantial improvement;
- GAN and VAE based methods such as StyleGAN and DeepSMOTE, do not have such a large effect on recall (although they help with general accuracy) which could be attributed to the data insufficiency of the minority classes leading to lower fidelity image generation.

Table 1: Results of data augmentation methods for balancing the PCL shavers dataset.

Method	Binary Recall	ROC-AUC
Resnet50	85.85	98.85
Resnet50 + SMOTE	95.84	98.87
Resnet50 + ADASYN	95.49	99.07
Custom CNN (see Figure 2)	95.84	99.20
Custom CNN + LW	96.07	99.09
StyleGAN	91.20	99.01
DeepSMOTE	93.58	99.23
Ours	97.27	99.34

Figure 3 shows a sample of the generated images from GAN and VAE-based methods across the three different classes. We distinguish between mild and pronounced defects according to how easily defects can be perceived visually by a human. All methods can generate images of sufficient quality, however generally StyleGAN images tended to produce small deviations close to a class average, while DeepSMOTE had some trouble reproducing small defects in the interrupted class. The BigGAN adaptation-based approach used in our method performed reasonably well but is also limited in producing small deviations around the existing images. For this reason, confidence awareness was introduced to make these small differences count more.

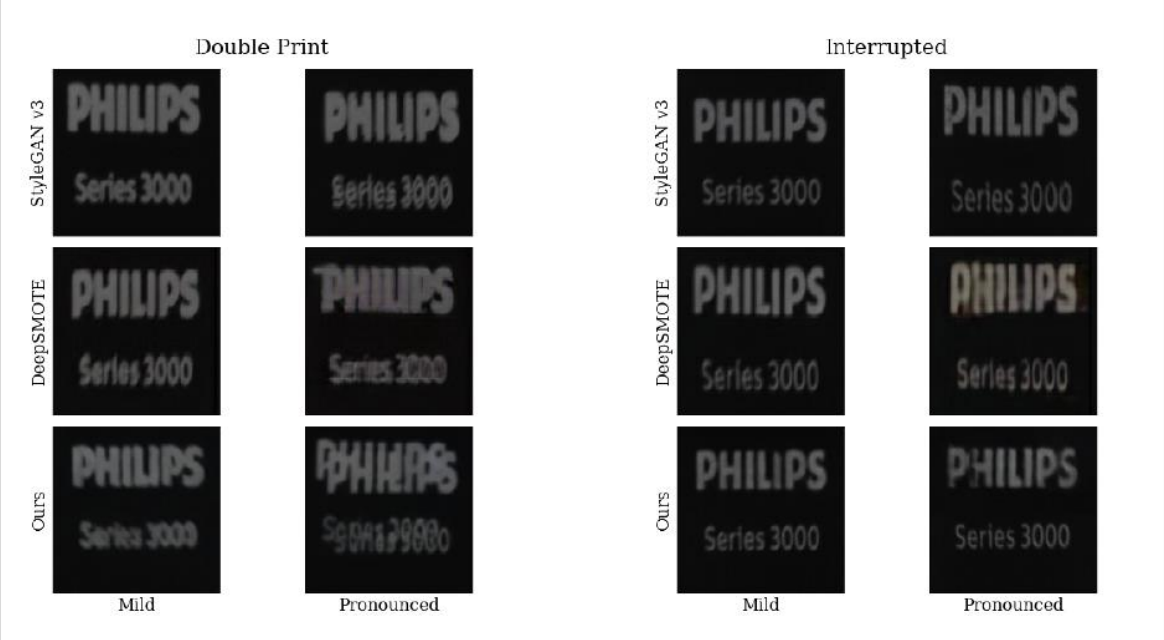


Figure 3. Synthetically generated defect images of different defect intensities

To guarantee uniformity for the experiments over all methods, the training epochs for the classifier were 50 with early stopping, so that all comparison classifiers have time to reach their loss plateaus. In terms of computation time, generation approaches such as StyleGAN and DeepSMOTE took the longest (20+ hours to train the generator) as StyleGAN's finetuning is heavy due to its sheer number of weights while DeepSMOTE needed to be trained from scratch to learn the linear interpolations. Our on-the-fly approach was substantially faster with 30 minutes needed for a full run; however, it was still slower than vector-based oversampling as BigGAN adaptation requires many weight updates as well.

3 Assessing and Enhancing Classifier Robustness through Simulated Novel Inputs

This part of our work is reflected in more detail in [REF-02] and [REF-03] and its implementation can be found at https://github.com/tspyrosk/osr-data-augmentation.

3.1 Background

It is important for all AI-decision making that operates in a real-life dynamic environment such as the manufacturing shop floor to be resilient to unexpected inputs and not limited to reacting only to the data it has been trained on. This resilience will prevent the AI from making misguided decisions that could harm human operators or put the integrity of the production process at risk. Although the main application of this task was the anticipation of unexpected defects during the PCL use case of Visual Quality Inspection, its applicability extends to scenarios where decisions are based on an AI classification (e.g., robotic perception).

Traditionally supervised AI classifiers learn to distinguish between different predefined classes, thus classifying any input, even when irrelevant to the problem at hand, as an instance of the known classes. In a practical setting this could be risky as there is a high probability of unknown inputs occurring ("unknown unknowns" [REF-20]) and consequently being treated as known. Unsupervised learning methods usually operate in a more "open" setting, however, their performance lags behind in problems where training data is readily available. Taking these two factors into account we identified two branches of recent AI research that develop systems robust to out-of-distribution samples, namely Open Set Recognition (OSR) and Semi-supervised Anomaly Detection:

- OSR is an attempt to minimize open-space risk (i.e., risk that an unknown sample will be classified into one of the known classes), while retaining performance on the closed-set (i.e., instances of the classes the AI has been trained on). This, as our experiments also show, usually requires a trade-off between open- and closed-set performance, which is determined differently by each classification algorithm.
- Semi-supervised approaches to defect detection require to be trained only on the non-defective class of products. Their goal is to learn the distribution of non-defective images and punish deviations with regard to predefined or adaptive thresholds. Though these methods are useful when too few images of the defect classes are available during training, they lag behind supervised learning methods that can learn defect classes even under substantial data imbalances. As our experiments show these achieve comparable open-set performance with supervised OSR methods but usually perform worse on the closed set.

The methods we compared were divided into three categories according to their implementation requirements. The first group foresees methods operating on vector data for which we used pre-extracted features from CNNs trained initially on Imagenet (Resnet50, VGG '16, and Inception v3). They include the baseline Multi-Layer Perceptron (MLP) followed by one-class classifiers: One-Class Support Vector Machines (One-class SVM), Isolation Forest, and Local Outlier Factor. Next, we have some of the SVM-based Open-set classifiers, namely Probability of Inclusion SVM (PI-SVM) and Weibull SVM (W-SVM), and finally, OpenMax, which is used to calibrate scores extracted from an MLP similar to the baseline. The second group consists of semi-supervised methods that learn only from the non-defective data, namely Ganomaly, Deep Feature Kernel Density Estimation (DFKDE), and Deep Feature Modelling

(DFM). The third group includes data augmentation techniques: Open Set Recognition with Counterfactual Images (OSRCI) and OpenGAN. Table 2 describes how each of these methods has been applied to our use case:

Table 2. Semi-supervised and Open Set Recognition Methods.

Method	Description
MLP	Single hidden layer architecture with 100 neurons leading to a 3-class classification head, both for the open and closed-set cases, with 'adam' optimizer.
One-class SVM	Despite being categorized as an unsupervised one-class classification method, OCSVM allows a small proportion of outlier instances in training, corresponding to the parameter nu. We fill out this proportion using the known defect instances in the training set. Otherwise, OCSVM works like a usual SVM but only forms a boundary for separating the good class from the rest of the instances. In our experiments, we used nu=0.3 and an RBF kernel.
Isolation Forest	The concept behind isolation forests is the linear splitting of the feature space by individual trees until a point is "isolated" in a tree leaf. The anomaly score assigned by the forest is an accumulation of how quickly each tree manages to separate the anomaly from the rest of the dataset. For the training of the IF we additionally use closed-set defect samples similar to OCSVM, setting the contamination factor (which again corresponds to the proportion of total defects to good images equal to 0.3). We also set the number of isolation tree estimators to 100.
Local Outlier Factor	A density-based anomaly detection method that is again trained on both the good and defect classes using a contamination factor of 0.3. The main idea behind LOF is that it compares the local point density of a given point to that of k of its neighbors and labels those with lower relative densities as anomalies. We chose k=20 neighbors based on the Euclidean distance.

WSVM [REF-21]	W-SVM is an ensemble of one-class and multi-class SVMs, whose scores are combined and calibrated using the Weibull distribution according to Extreme Value Theory (EVT). The ultimate goal is to flatten class probabilities in the open space to 0. We used an RBF kernel and a 0.1 probability threshold for rejecting samples as open set.
PI-SVM [REF-22]	A more sophisticated extension of W-SVM is trying to model the probability of inclusion for each class using only in-class samples and EVT. The model was parameterized in a similar way to W-SVM.
OpenMax [REF-23]	OpenMax operates on the penultimate layer of a DNN to accommodate an "unknown" class and recalibrates scores using EVT. We used a tail size of ten samples to fit the Weibull distribution and an alpha = 3 corresponding to the total number of classes whose scores are recalibrated. In our case, we have very few (three) original classes, so we recalibrate all of them. For the DNN, we use the same MLP on top of pre-trained embeddings as above.
Ganomaly [REF-24]	GANomaly is based on an encoder-decoder-encoder architecture, where the starting encoder-decoder pair is trained similarly to DCGAN. To measure how anomalous an image is, the difference between the final output and the first encoder's latent output is taken into account. We use a latent vector size of 100 dimensions along with w_adv = 1, w_con = 50, and w_enc = 1 for the coefficients of the adversarial, contextual and encoder loss coefficient defined in [REF-24].
DFKDE [REF-25]	This method consists of a backbone network to extract deep features followed by Principal Component Analysis (PCA) and Gaussian Kernel Density Estimation. In our use case, we use the 16 principal components explaining the most variance along with the Euclidean distance and a 0.5 score threshold for anomaly classification.

DFM [REF-26]	This approach tries to fit a Gaussian distribution or mixture of Gaussians to a DNN's features after a DNN has been trained on a specific classification task and PCA has been applied to the feature vectors to reduce their dimensionality and thus improve computational speed. In our use case, we train the model only on good images and use a Resnet50 backbone with a 0.97 variance retaining threshold for the PCA and the feature reconstruction score to rank anomalies.
OSRCI [REF-27]	A data augmentation technique that tries to produce counterfactual augmentation images by searching for the nearest GAN latent input that leads to misclassification of the generated image. In terms of parameters, we followed using a 20-dimensional latent space and a classifier architecture of two convolutional layers followed by two fully connected layers.
OpenGAN [REF-28]	Slightly different from OSRCI, OpenGAN uses latent space interpolations between classes to produce the augmentation images combined with semantic information from features extracted per input sample. A Resnet18 backbone is used for the feature extractor and Gaussian Kernel Density estimation for the final classifier.

3.2 Methods

Following in the vein of OpenGAN and OSRCI we also attempted at simulating novel inputs (defects) and using them as an extra class to augment the training dataset of product images. Our method is also based on GAN manipulation for which we rely on one of the most sophisticated and expressive networks available, namely StyleGAN v3. Aside from being able to produce very high-quality synthetic images, the StyleGAN family of models also utilized latent spaces that are highly manipulable, thus giving the user some degree of control over the generation process, which we will utilize as seen below.

As shown in Figure 4, our approach consists of three basic components, namely the *Generation of Synthetic Data*, the *Voting-based Filtering* and the *Training on the Augmented Dataset.*

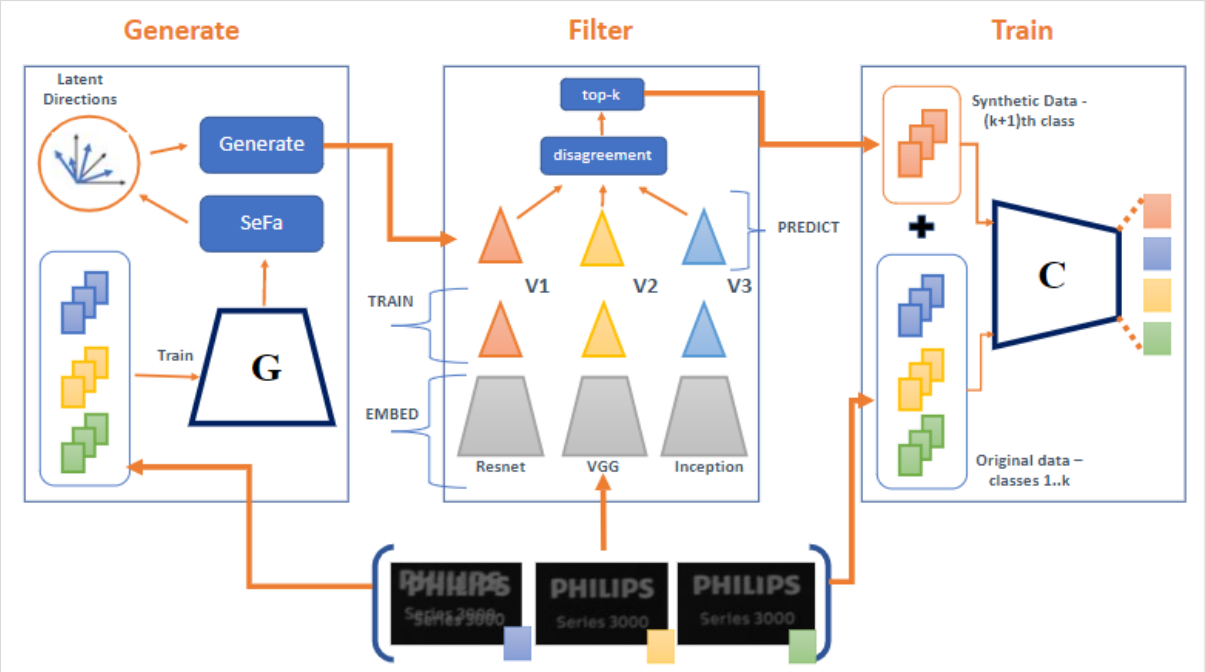


Figure 4. Process for generating synthetic augmentation data to increase robustness to novel inputs.

To perform data generation, we used a StyleGAN v3 model [REF-18] (labeled as G in Figure 4) pre-trained on the CelebHQ dataset of human faces and fine-tuned on the original PCL shaver dataset. Our aim thereafter was to control StyleGAN generation so that we generate novel inputs for this class. This is made possible due to the generalization capabilities of StyleGAN which allow generation to sometimes extend the limits of the input distribution. Our aim is then distilled to producing these images that bound the distribution of each class by being just a little outside of it. To make this process efficient we rely on the Semantic Latent Factorization (SeFa) module introduced in [REF-29], which discovers directions of maximal output change in StyleGAN's latent space. While traversing these directions we collect a variety of candidate images, that after filtering by the subsequent component might be the boundary images we are looking for.

To filter these generated images, we need to determine what a distribution-boundary image looks like. Our definition of such an image is one that causes high confusion to a set of voter classifiers. This is because classifiers with different architectures will all learn in a similar way when it comes to in-distribution images. However, in the "open space" where classification boundaries are not data-determined and therefore more random, each classifier will learn different boundaries causing classifiers to disagree when given an unexpected out-of-distribution point. This is exactly the criterion we use, i.e., the number of different class assignments of an image over a set of voter classifiers with different architectures. These classifiers are all trained on the same training set as the GAN and the final classifier.

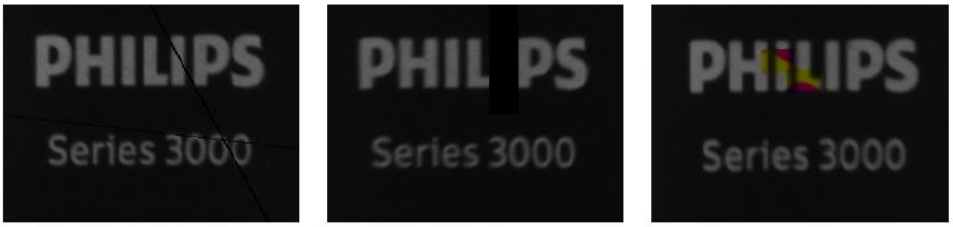
Finally, filtered images have added an extra augmentation class to the training dataset representing "unknown" inputs. In our experiment, a classifier trained on this augmented dataset shows very promising results in both the open- and closed-set classes.

3.3 Evaluation

To evaluate all these methods, we synthetically generated 5 novel classes of images, by graphically simulating possible unexpected defects that could occur during the inspection process:

- **Line Interruptions**, which could result from pre-existing scratches on the printing pad;
- **Missing Letters**, which could be due to a defect in the printer head;
- **Discoloration**, due to the corruption/mixing of the sprayed colour;
- **Horizontal Flips** due to a wrong setting of the printer head;
- **Vertical Flips** due to a wrong setting of the printer head in the vertical direction;

Examples of the first three categories are shown in Figure 4. Images from these categories are merged with the test set so that the test set contains 800 good images and 250 images with known defects (double prints and interrupted) split as approx. 30% of the original dataset and 250 unknown defect images - 50 per synthetic class.



(a) Line Scratches (b) Missing Letter (c) Discoloured

Figure 5. Artificially generated defect classes (open-set) used for testing.

Four metrics are used to compare our method on the aforementioned test set:

- The Area Under the Receiving Operating Characteristic (AUROC) curve;
- The F1-Score;
- The Binary Recalls from the perspective of the defect class for closed-set and open-set defects.

To keep the evaluation consistent between OSR and semi-supervised methods, we chose binary metrics that do not distinguish between specific defect classes. This is also in alignment with our use case, where, in order to be examined by human operators before being discarded, samples are identified as either "defects" or "unknown" corresponding to whether the given sample is OK or needs to be manually checked by an operator. For this reason, emphasis is given to the recall metric for defect classes which indicates what percentage of defects are automatically held back from moving through the system unnoticed in case they are mistakenly marked as good. This metric is also refined to distinguish between open and closed set classes and allow us to examine potential trade-offs between performance on known and unknown types of defects. Finally, the F1-score and AUROC metrics are used as safety valves to guarantee that the models achieve reasonable prediction performance in the good class. As mentioned previously, there is a danger that some methods might show perfect recall at the cost of marking many more images than necessary as defects. This would make the automated quality inspection inefficient for a practical setting, owing to the increased burden on human operators, especially given that defective products are a tiny minority in comparison to the total that passes through the system. If we were to choose a single most balanced

metric for comparison, that would be the F1 metric as it considers both open- and closed-set performance information with reduced imbalance bias.

The results are the average outcomes of 30 independently seeded runs and were executed in an environment with 4 CPU cores of 2.3GHz, 16GB of RAM, and access to an NVidia K80 GPU.

Table 3, Table 4 and Table 5 show the results for methods operating on pre-extracted features derived from three base networks. Some interesting observations we can arrive at are the following:

- Most methods perform on top of VGG embeddings (e.g., MLP, PI-SVM, W-SVM, and OpenMax).
- Inception embeddings with PI-SVM and OpenMax are also good combinations.
- It is evident that Resnet50 features lead most methods to perform significantly worse, especially on the open-set problem.
- One-class classifiers (one-class SVM, Isolation Forests, Local Outlier Factor) seem to achieve perfect recall on open-set instances irrespective of the underlying embeddings. However, this comes with a significant decrease in closed-set recall highlighting the trad-off mentioned in the Background section.
- Surprisingly, the baseline MLP method with VGG embeddings performs extremely well without using open-set mechanisms, which could be attributed to the richness of VGG's extracted features.
- Our proposed data augmentation method shows consistency across different embeddings (even Resnet50) and maintains high closed- and open-set recall scores.

Table 3. Results of OSR methods using Resnet50 embeddings.

Method	AUROC	F1	Recall Closed-Set	Recall Open-Set
MLP	0,7414	0,8462	0,8800	0,2386
One-class SVM	0,8353	0,7924	0,6245	1,0000
Isolation Forest	0,8764	0,8213	0,6920	1,0000
Local Outlier Factor	0,9121	0,8451	0,7285	1,0000
WSVM	0,8596	0,7659	0,7385	0,8274
PI-SVM	0,6709	0,8375	0,8628	0,1684
OpenMax	0,6973	0,7737	0,9165	0,5834
Ours	0,9952	0,9650	0,8670	0,9772

Table 4. Results of OSR methods using VGG '16 embeddings.

Method	AUROC	F1	Recall Closed-Set	Recall Open-Set
MLP	0,9777	0,9633	0,9320	0,9208
One-class SVM	0,8767	0,8256	0,7060	1,0000
Isolation Forest	0,8731	0,8598	0,8607	0,8664
Local Outlier Factor	0,9090	0,8430	0,7095	1,0000
WSVM	0,9022	0,8088	0,8122	0,9631
PI-SVM	0,9902	0,9533	0,9111	1,0000
OpenMax	0,9630	0,9389	0,9707	0,8932
Ours	0,9965	0,9796	0,9560	0,9952

Table 5. Results of OSR methods using Inception v3 embeddings.

Method	AUROC	F1	Recall Closed-Set	Recall Open-Set
MLP	0,9325	0,9117	0,8695	0,6754
One-class SVM	0,8771	0,8258	0,7080	1,0000
Isolation Forest	0,9051	0,8389	0,7565	1,0000
Local Outlier Factor	0,9149	0,8498	0,7030	1,0000
WSVM	0,9106	0,6797	0,9200	0,8856
PI-SVM	0,9834	0,9577	0,9040	0,9856
OpenMax	0,9409	0,9289	0,9500	0,8464
Ours	0,9954	0,9752	0,9490	0,9884

Table 6 compares semi-supervised and data-augmentation-based methods. Since semi-supervised methods are trained only on the non-defective portion of the dataset, all defect classes can be considered open-set as they are unknown at training time. Nevertheless, we keep the separate evaluations for open- and closed-set as they highlight a potential deficit of this family of methods, namely the fact that by design they cannot take advantage of known defect instances. DFM has the best performance across all metrics and tends to outperform data augmentation methods, something that we attribute to the lower fidelity output of older GAN architecture in conjunction with the high similarity between defects and non-defects. Despite semi-supervised methods achieving a lower recall than methods from Tables 3-5 on

the closed-set classes, they manage to obtain high AUROC scores, possibly due to their better-calibrated probability outputs.

Table 6. Results of Semi-supervised and Data Augmentation methods.

Method	AUROC	F1	Recall Closed-Set	Recall Open-Set
Ganomaly	0,8242	0,8930	0,6100	0,9460
DFKDE	0,9848	0,7401	0,7700	0,9720
DFM	0,9909	0,8347	0,8500	0,9800
OSRCI	0,7884	0,6813	0,9900	0,8540
OpenGAN	0,9399	0,8858	0,8483	0,6860
Ours + VGG	0,9965	0,9796	0,9560	0,9952

As for data augmentation methods specifically, OSRCI, shows high defect recall scores with lower AUROC and F1, hinting at a potential marking of many good instances as defects. On the other hand, OpenGAN is more stable across these metrics despite slightly lower recalls. Our method's improvement upon them could be attributed in part to StyleGAN's higher expressive and generalization capabilities compared to earlier GAN architectures, but also to our novel filtering mechanism.

4 Conclusion

In this deliverable, we outline and evaluate a prototype of the Simulated Reality consisting of two main functionalities that address important issues when it comes to AI-safety and reliability and the application of AI in practical real-life environments, namely the resilience to data imbalance and the resilience to novel unexpected inputs. As per the name of our component, our approach in both cases was one of Data Augmentation through the simulation of input data. This cuts at the heart of modern AI systems which leverage the availability of data to enable complex decision-making and is aligned with recent approaches in the field of data-driven AI, according to which it might be more effective and practical to improve the quality of the input data rather than improving the algorithm itself. Both functionalities are intended to be applied to the PCL Visual Quality Inspection pilot, by directly addressing the two identified issues of the relative lack of defect data and the possibility of previously unanticipated production defects

Our plans for the next steps towards the deployment of the STAR components are to adjust the applied methods in deployable interfaces so that they integrate with the WP4 architecture and are easily usable by other components such as the Active Learning Component. We will also extend them as much as possible to fit other datasets and use cases.

References

Reference	Name of document
[REF-01]	Spyros Theodoropoulos, Patrik Zajec, Joze M. Rozanec, Dimosthenis Kyriazis and Panayiotis Tsanakas. On-the-fly Image-level Oversampling for Imbalanced Datasets of Manufacturing Defects. Submitted to Machine Learning: Special Edition on Imbalanced Learning, Springer, 2022.
[REF-02]	Spyros Theodoropoulos, Patrik Zajec, Joze M. Rozanec, Dimitrios Dardanis, Georgios Makridis, Dimosthenis Kyriazis and Panayiotis Tsanakas. Identifying Novel Defects during AI-driven Visual Quality Inspection. Accepted at IFAC 2023.
[REF-03]	Spyros Theodoropoulos, Dimitrios Dardanis, Georgios Makridis, Patrik Zajec, Joze M. Rozanec, Dimosthenis Kyriazis and Panayiotis Tsanakas. Enhancing Robustness to Novel Visual Defects through StyleGAN Latent Space Navigation: A Manufacturing Use Case. Submitted to Journal of Intelligent Manufacturing, 2023.
[REF-04]	Fathy, Y., Jaber, M., Brintrup, A.: Learning with imbalanced data in smart manufacturing: A comparative analysis. IEEE Access 9, 2734-2757 (2021).
[REF-05]	See, J.E.: Visual inspection : a review of the literature. Sandia Report SAND2012-8590, Sandia National Laboratories, Albuquerque, New Mexico (2012)
[REF-06]	Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016).
[REF-07]	Zhang, G., Cui, K., Hung, T.-Y., Lu, S.: Defect-gan: High- delity defect synthesis for automated defect inspection. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 2523-2533 (2021).
[REF-08]	Saiz, F.A., Alfaro, G., Barandiaran, I., Grana, M.: Generative adversarial networks to improve the robustness of visual defect segmentation by semantic networks in manufacturing components. Applied Sciences 11(14) (2021).
[REF-09]	Pawara, P., Okafor, E., Schomaker, L., Wiering, M.: Data augmentation for plant classi cation. In: Blanc-Talon, J., Penne, R., Philips, W., Popescu, D., Scheunders, P. (eds.) Advanced Concepts for Intelligent Vision Systems, pp. 615-626. Springer, Cham (2017)
[REF-10]	Yun, J.P., Shin, W.C., Koo, G., Kim, M.S., Lee, C., Lee, S.J.: Automated defect inspection system for metal surfaces based on deep learning and data augmentation. Journal of Manufacturing Systems 55, 317-324 (2020).
[REF-11]	Sampath, V., Maurtua, I., Aguilar Marten, J.J., Gutierrez, A.: A survey on generative adversarial networks for imbalance problems in computer vision tasks. Journal of Big Data 8, 12 (2021).
[REF-12]	Luan, F., Paris, S., Shechtman, E., Bala, K.: Deep painterly harmonization. Computer Graphics Forum 37 (2018).

[REF-13]	Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. J. Artif. Int. Res. 16(1), 321-357 (2002)
[REF-14]	Han, H., Wang, W., Mao, B.: Borderline-smote: A new over-sampling method in imbalanced data sets learning. In: ICIC (2005)
[REF-15]	He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), pp. 1322-1328 (2008).
[REF-16]	Noguchi, A., Harada, T.: Image generation from small datasets via batch statistics adaptation. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2750-2758 (2019)
[REF-17]	Elsayed, G.F., Krishnan, D., Mobahi, H., Regan, K., Bengio, S.: Large margin deep networks for classification. (2018). https://arxiv.org/pdf/1803.05598.pdf
[REF-18]	Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. IEEE Transactions on Pattern Analysis & Machine Intelligence 43(12), 4217-4228 (2021).
[REF-19]	Dablain, D., Krawczyk, B., Chawla, N.V.: Deepsmote: Fusing deep learning and smote for imbalanced data. IEEE Transactions on Neural Networks and Learning Systems, 1-15 (2022).
[REF-20]	Dietterich, T.G. (2017). Steps toward robust artificial intelligence. AI Magazine, 38(3), 3–24.
[REF-21]	Scheirer, W.J., Jain, L.P., Boult, T.E.: Probability models for open set recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 36(11), 2317–2324 (2014).
[REF-22]	Jain, L.P., Scheirer, W.J., Boult, T.E.: Multi-class open set recognition using probability of inclusion. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014, pp. 393–409. Springer Cham (2014)
[REF-23]	Rozsa, A., Gunther, M., Boult, T.E.: Adversarial robustness: Softmax versus openmax. ArXiv abs/1708.01697 (2017)
[REF-24]	Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semisupervised anomaly detection via adversarial training. In: Jawahar, C.V., Li, H., Mori, G., Schindler, K. (eds.) Computer Vision – ACCV 2018, pp. 622–637. Springer, Cham (2019)
[REF-25]	Akcay, S., Ameln, D., Vaidya, A., Lakshmanan, B., Ahuja, N., Genc, U.: Anomalib: A Deep Learning Library for Anomaly Detection (2022)
[REF-26]	Ahuja, N.A., Ndiour, I., Kalyanpur, T., and Tickoo, O. (2019). Probabilistic modeling of deep features for out-of-distribution and adversarial detection. doi:10.48550/ARXIV.1909.11786. URL https://arxiv.org/abs/1909.11786.

[REF-27]	Neal, L., Olson, M., Fern, X., Wong, W.-K., Li, F.: Open set learning with counterfactual images. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
[REF-28]	Ditria, L., Meyer, B.J., Drummond, T.: Opengan: Open set generative adversarial networks. In: ACCV (2020)
[REF-29]	Shen, Y., Zhou, B.: Closed-form factorization of latent semantics in gans. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1532–1540. IEEE Computer Society, Los Alamitos, CA, USA (2021).