Data Augmentation via Mixed Class Interpolation using Cycle-Consistent Generative Adversarial Networks Applied to Cross-Domain Imagery

Hiroshi Sasaki[1], Chris G. Willcocks[1], Toby P. Breckon[1,2]
Department of {[1]Computer Science | [2]Engineering}, Durham University, Durham, UK

***Abstract*—Machine learning driven object detection and classification within non-visible imagery has an important role in many fields such as night vision, all-weather surveillance and aviation security. However, such applications often suffer due to the limited quantity and variety of non-visible spectral domain imagery, in contrast to the high data availability of visible-band imagery that readily enables contemporary deep learning driven detection and classification approaches. To address this problem, this paper proposes and evaluates a novel data augmentation approach that leverages the more readily available visible-band imagery via a generative domain transfer model. The model can synthesise large volumes of non-visible domain imagery by image-to-image (I2I) translation from the visible image domain. Furthermore, we show that the generation of interpolated mixed class (non-visible domain) image examples via our novel Conditional CycleGAN Mixup Augmentation (C2GMA) methodology can lead to a significant improvement in the quality of non-visible domain classification tasks that otherwise suffer due to limited data availability. Focusing on classification within the Synthetic Aperture Radar (SAR) domain, our approach is evaluated on a variation of the Statoil/C-CORE Iceberg Classifier Challenge dataset and achieves 75.4% accuracy, demonstrating a significant improvement when compared against traditional data augmentation strategies (Rotation, Mixup, and MixCycleGAN).**

I. Introduction

The demand for automated pattern recognition, especially automatic object detection and classification in imagery, is continuously expanding. In computer vision, there are many applications utilising automatic pattern recognition, for example, optical character recognition [1], video surveillance [2], agricultural analysis from satellite imagery [3], and defect detection in factory automation [4]. These functions are enabled by recent advances in machine learning, namely deep neural networks (DNN) [5]. DNN have enabled hitherto unprecedented performance on various challenging computer vision tasks such as image classification, object detection, semantic segmentation and temporal video analysis.

This expansion, both in demand and performance, has led to the broader consideration of computer vision applications in imagery domains beyond the visible spectrum, i.e. non-visible images such as infrared (thermal) [6], synthetic aperture radar (SAR) [7] and X-ray images [8]. Imaging within the non-visible spectrum provides sensing capabilities ranging from all-weather visibility, object temperature, material characteristics and sub-surface/object transparency. Whilst

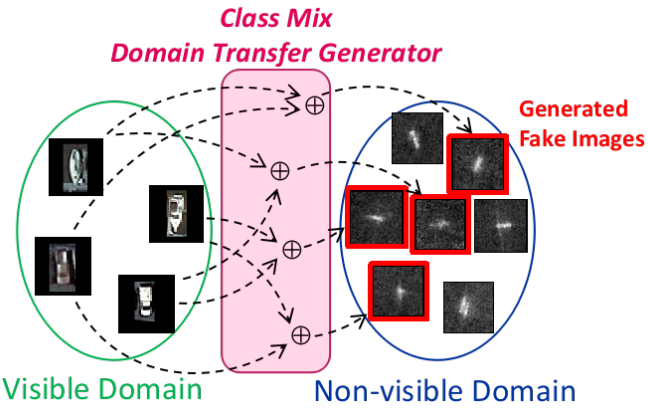


Fig. 1: Conceptual illustration of our novel data augmentation approach for generating cross-domain, class-interpolated image instances.

DNN approaches have predominately been applied to visible domain imagery, they are readily applied across the non-visible spectrum. However, the primary challenge is the low data availability in these additional spectral imaging domains. Whilst contemporary DNN approaches generally perform well in domains with large amounts of data available, within the non-visible imaging domain data availability is often more limited and it can be difficult to collect enough image samples to provide sufficient variability and coverage of the target data distribution expected at inference (test, deployment) time. For example, SAR imagery is far less readily available and accessible due to both the lesser prevalence of this sensing technology and its associated costs. In addition, SAR imagery substantially differs from visible-band imagery because it results from active sensing by microwave radar backscatter projection, whilst visible images are captured passively according to the intensity of reflected scene illumination. Moreover, SAR imagery is significantly impacted by the choice of microwave bands in use and by the angle of microwave transmission. These variations from conventional imagery that preclude the direct applicability of commonplace transfer learning solutions, coupled with the lack of data availability, further inhibit inter-task applications with such

diverse sensor imagery.

In order to address this issue of DNN model generalisation under such limited data availability, data augmentation methods such as geometric image transformation and pixel-wise intensity transformations are traditionally adopted. However, such methods tend to synthesise images which are highly biased to both the prior assumptions of this augmentation and the prior distribution of the already limited dataset in use. An alternative solution, more specific to object classification tasks, involves blending a pair of input images of different classes to smooth the classification decision boundary during the training [9]. This approach can be effective when there are few training examples (limited data availability), but remains highly sensitive to biases in the input samples. To overcome these issues, recent research into image synthesis and dataset augmentation has focused on stochastic generative models, which can create a variety of high-quality images [10]. In particular, image-to-image (I2I) translation models are able to generate samples by mapping between image domains [11], whereas standard generative models synthesise images by transforming random values sampled from a simpler prior distribution. I2I translation is particularly effective when there are few images in a desired domain and large quantities of data available in another indirectly related domain, such as in the context of SAR images and publicly available visible images.

Taking this into consideration, we exploit the potential of I2I translation as a dataset augmentation strategy and develop a new I2I translation model, adopted from Cycle-Consistent Generative Adversarial Networks (CycleGAN) [11]. In particular, we modify CycleGAN by manipulating class conditional information and generating class-interpolated images (Figure 1), as described in detail in Section III. The experiments supporting our method, within the context of SAR object classification, are presented in Section IV with subsequent conclusions presented in Section V.

II. Related Work

Many data augmentation approaches within a computer vision context have been proposed and divided into two sub-types: unsupervised and supervised [12].

A. Unsupervised Data Augmentation

An unsupervised approach aims to increase the quantity of training imagery via a set of fixed geometric and pixel-wise image processing operations to transform an existing dataset image (e.g. flipping, rotation, cropping, adding noise, etc. [12]).

Mixup [9] is a recent approach that blends pairs of randomly chosen training images using randomly weighted blending rates to avoid overfitting. In addition, [13] [14] [15] [16] have shown the effectiveness of partially masking image sub-regions to force generalisation during model training. Instead of zero maskings, CutMix [17] replaces these regions with a region of the same size from another training set image and provides an improvement in performance.

B. Supervised Data Augmentation

While unsupervised methods can reduce overfitting, the trained models are often unable to accurately model patterns or trends that appear within the test distribution that are infrequent within the training data distribution. This is largely due to the fact that unsupervised augmentation approaches transform data sampled from the same underlying training distribution, therefore their outputs reflect the inherent biases and patterns in this original training distribution. In order to overcome this issue, several supervised approaches have been proposed that instead generate new images using additional label information to improve generalisation between domains [18] [19].

Manifold Mixup [18] is a modification of Mixup. This interpolates not only input images and their associated output labels but also latent information within the hidden layers. This attempts to increase the novelty of data samples generated by latent information level processing. Meanwhile, data augmentation via diversification of image style was proposed [19]. Utilising a style transfer network [20], a DNN trained to transfer the style from one image to another while preserving its semantic content, they additionally augmented their training data via image style randomisation.

Generative Adversarial Networks (GAN) [10] have significantly impacted data augmentation within DNN training. GAN is a generative DNN architecture, designed to have a generator and a discriminator component that compete against each other during its training process. The generator is trained to map randomised values to real data examples by the discriminator output. The discriminator is simultaneously trained to discriminate real and fake data examples produced by the generator. The objective function is defined as:

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

where G and D are the generator and discriminator respectively. x is input data and z is random noise. As a result of training the generator within this GAN architecture, it is hence optimised to create realistic, yet artificial data that is statistically similar (drawn from the same distribution) as the real data. The GAN architecture has been shown to be effective with convolutional neural networks, popularised by Deep Convolutional GAN (DCGAN) [21]. A Conditional GAN (cGAN) [22] was proposed to modify the GAN architecture to take account of classes by adding class labels into the inputs of the generator and discriminator. The objective function (1) is modified as:

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

where y is the category label given in the objective function. Moreover, another GAN variant, called Auxiliary Classifier GAN (ACGAN) [23], implemented classification in addition to generative modelling. This architecture trains its network

to minimise the distance of between both the real and fake data examples and the actual and predicted category labels. While such conditional information was initially implemented as a concatenation of the input and output of the networks, other methods find that the conditional information can be incorporated into the normalisation layers to significantly improve the generated results [24] [25]. These normalisation layers are called the conditional normalisation layers and the generators are modified as $G(z, e(y))$, where e is the embedding function. The effectiveness of this conditional label embedding has been not only been used in the generator, but also to the discriminator. This 'projection discriminator' is implemented by an inner product of the embedded one-hot labels and the intermediate layer outputs [26].

A large corpus of images from other related domains can also be useful for increasing training data in some cases. Generating new images by transferring from another domain image set, which is called I2I translation, has the possibility of expanding the distribution of training data such that it retains more of the structure of real images rather than the synthesised images generated only from vectors of noise. CycleGAN [11] is one of the expansions of GAN specified in I2I translation. In this method, G and D are trained to transfer from source images $x_s \in X_s$ to target images $x_t \in X_t$. This not only learns a lateral transform, but also the bilateral transform paths $G_t(x_s), G_s(x_t)$. In addition, this adopts a new loss measure named a cycle-consistency loss $L_{\text{cyc}}(G_s, G_t)$, which is represented as:

$$L_{\text{cyc}}(G_s, G_t) = \mathbb{E}_{x_s \in X_s}[\|G_s(G_t(x_s)) - x_s\|_1] + \mathbb{E}_{x_t \in X_t}[\|G_t(G_s(x_t)) - x_t\|_1] \quad (3)$$

In total, the full objective function is:

$$\min_{G_s, G_t} \max_{D_s, D_t} V(D_s, G_s) + V(D_t, G_t) + \lambda_{\text{cyc}} L_{\text{cyc}}(G_s, G_t) \quad (4)$$

where λ_{cyc} is a cycle-consistency loss weight.

MixCycleGAN [27] applies a 'mixup' operation to the CycleGAN process to stabilise the training and increase the variety of the generated outputs. This method splits an input image into two rectangular regions vertically or horizontally and replaces one region with that of another image:

$$\begin{aligned} \bar{x} &= \text{cat}(x_1[:\lambda H, :], x_2[(1-\lambda)H :, :]) \\ &\text{or } \text{cat}(x_1[:, :\lambda W], x_2[:, (1-\lambda)W :]) \end{aligned} \quad (5)$$

where, $\bar{x}$ is the mixed image, $x_1, x_2 \in X$ are the input images, H, W are the height and width of the input images respectively, and cat is a concatenation function. $\lambda \in [0, 1]$ is the mixup ratio, and $\lambda \sim \text{Beta}(\alpha, \alpha)$ is from the beta distribution Beta, in which α is constantly set as in [9]. The preprocessed mixed image $\bar{x}$ is input to the generator G of CycleGAN to synthesise a fake image. The discriminator D is modified to estimate the mixup ratio from the alpha-blended real and fake images, which is optimised as:

$$\min \mathbb{E}_{x \in X}[\log |\lambda - D(\lambda x + (1-\lambda)G(\bar{x}))|] \quad (6)$$

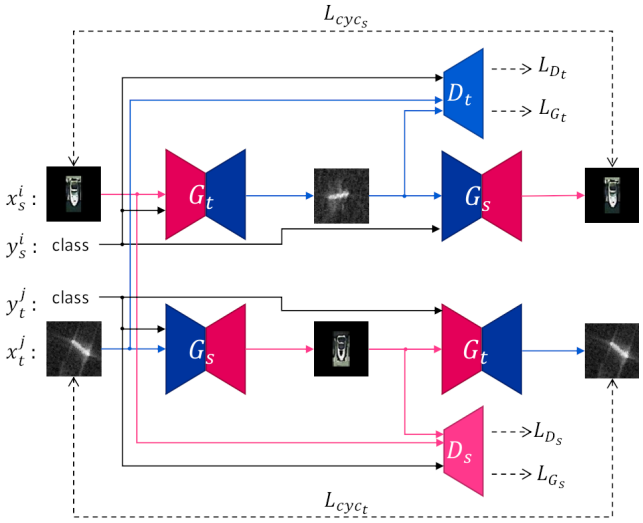


Fig. 2: Overall flow of our conditional CycleGAN model.

Our approach is similar to MixCycleGAN. However, while MixCycleGAN stitches rectangular image regions and does not use class labels, our approach adopts cGAN with the conditional normalisation layers and the projection discriminator to allow the class labels as input for the generator to enforce synthesising class-specific images. This proposed strategy enables generation of more sophisticated class-interpolated images by alpha-blending of the input images and class labels, rather than with a simple rectangular image region mixup. The details of these are described in Section III.

III. Methodology

The proposed method assumes a source domain dataset $(x_s^i, y_s^i) \in X_s^N$ and a target domain dataset $(x_t^j, y_t^j) \in X_t^M$ which consist of N and $M (\ll N)$ samples respectively. x_s^i and x_t^j are the images themselves and y_s^i and y_t^j are class labels. The types of classes are common in both domains.

Initially, a generative model, which transfers between two different domains, is built using the conditional CycleGAN approach. In order to prevent mode collapse and stabilise training, Spectral Normalization [28] is combined with the gradient penalty [29] as proposed in [30]. Furthermore, as discussed previously, we apply conditional regularisation of cGAN to our CycleGAN model by implementing conditional normalisation layers and projection discriminators to improve the output quality. The overall flow is shown in (Figure 2) where, unlike ordinary CycleGAN, the generator and discriminator functions are conditioned on the class labels. The objective function is defined as a simple sum of weighted terms:

$$L = \lambda_s L_{G_s} + \lambda_t L_{G_t} + \lambda_s L_{D_s} + \lambda_t L_{D_t} + \lambda_s \lambda_{\text{cyc}} L_{\text{cyc}_s} + \lambda_t \lambda_{\text{cyc}} L_{\text{cyc}_t} \quad (7)$$

where:

$$L_{G_s} = \mathbb{E}_{(x_t^j, y_t^j) \in X_t}[\log(1 - D_s(G_s(x_t^j, e_t(y_t^j)), e_s(y_t^j)))] \quad (8)$$

$$L_{G_t} = \mathbb{E}_{(x_s^i, y_s^i) \in X_s}[\log(1 - D_t(G_t(x_s^i, e_s(y_s^i)), e_t(y_s^i)))] \quad (9)$$

$$L_{D_s} = \mathbb{E}_{(x_s^i, y_s^i) \in X_s}[\log(1 - D_s(x_s^i, e_s(y_s^i)))] + \mathbb{E}_{(x_t^j, y_t^j) \in X_t}[\log(D_s(G_s(x_t^j, e_t(y_t^j)), e_s(y_t^j)))] + \lambda_{\text{gp}} \mathbb{E}_{(\hat{x}_s^j, \hat{y}_s^j) \sim \mathbb{P}_{\hat{x}_s, \hat{y}_s}}[(\|\nabla D_s(\hat{x}_s^j, e_s(\hat{y}_s^j))\|_2 - 1)] \quad (10)$$

$$L_{D_t} = \mathbb{E}_{(x_t^j, y_t^j) \in X_t}[\log(1 - D_t(x_t^j, e_t(y_t^j)))] + \mathbb{E}_{(x_s^i, y_s^i) \in X_s}[\log(D_t(G_t(x_s^i, e_s(y_s^i)), e_t(y_s^i)))] + \lambda_{\text{gp}} \mathbb{E}_{(\hat{x}_t^j, \hat{y}_t^j) \sim \mathbb{P}_{\hat{x}_t, \hat{y}_t}}[(\|\nabla D_t(\hat{x}_t^j, e_t(\hat{y}_t^j))\|_2 - 1)] \quad (11)$$

$$L_{\text{cyc}_s} = \mathbb{E}_{(x_s^i, y_s^i) \in X_s}[\|(G_s(G_t(x_s^i, e_s(y_s^i)), e_t(y_s^i)) - x_s^i)\|_1] \quad (12)$$

$$L_{\text{cyc}_t} = \mathbb{E}_{(x_t^j, y_t^j) \in X_t}[\|(G_t(G_s(x_t^j, e_t(y_t^j)), e_s(y_t^j)) - x_t^j)\|_1] \quad (13)$$

λ_s and λ_t are source domain and target domain weights, respectively. λ_{gp} is a weight of the gradient penalty. That is, we balance the corresponding generator and discriminator functions with the cycle-consistency losses for both the source and target domains accordingly.

After training, the model is used for the synthesis of new class-conditioned images via the domain transfer. A pair of images and class labels in the source domain dataset $(x_s^i, y_s^i), (x_s^j, y_s^j) \in X_s^N$ are used as an input. Subsequently, the input is processed to produce a tuple of a mixed image, label, and embedded feature vector $(\bar{x}_s^k, \bar{y}_s^k, \bar{e}_s^k)$, defined by:

$$\bar{x}_s^k = x_s^i * \lambda + x_s^j * (1 - \lambda) \quad (14)$$

$$\bar{y}_s^k = y_s^i * \lambda + y_s^j * (1 - \lambda) \quad (15)$$

$$\bar{e}_s^k = e_s(y_s^i) * \lambda + e_s(y_s^j) * (1 - \lambda) \quad (16)$$

where $\lambda \in [0, 1]$ is the mixup ratio, and $\lambda \sim \text{Beta}(\alpha, \alpha)$ from the beta distribution Beta, in which α is constantly set as in [9]. As a result, the mixed pair $(\tilde{x}_t^k, \tilde{y}_t^k)$ that is input to the generator and discriminator is defined, where:

$$(\tilde{x}_t^k, \tilde{y}_t^k) = (G_t(\bar{x}_s^k, \bar{e}_s^k), \bar{y}_s^k) \quad (17)$$

As a result, N samples $\tilde{X}_t^N = \{(\tilde{x}_t^k, \tilde{y}_t^k)\}$ are synthesised. The new fake samples are combined with the original dataset as $X_t^M \cup \tilde{X}_t^N$, where we denote this method as Conditional CycleGAN Mixup Augmentation (C2GMA).

IV. Experiments

The method is evaluated in the context of the ships/icebergs SAR classification task using the Statoil/C-CORE Iceberg Classifier Challenge dataset [31]. Results are compared between classification models trained with and without existing dataset augmentation approaches in addition to our proposed CycleGAN driven C2GMA (Section III) approaches.

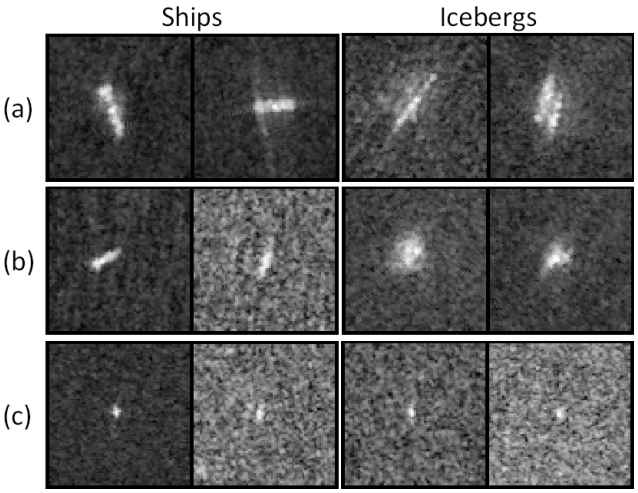


Fig. 3: SAR ships/icebergs images divided into three groups based on difficulty of discrimination by distance, angle, object size, etc.

A. Dataset

The Statoil/C-CORE Iceberg Classifier Challenge dataset [31] has a collection of satellite SAR images of ships and icebergs, each with 75×75 pixels. The dataset comprises of a training set with images labelled as either a ship or an iceberg, alongside a set of unlabelled test images. We use only the labelled training data in our experiments (we split this labelled data into different groups for evaluation, discussed subsequently). Each sample in the data is represented by 2-channel floating-point images according to the two different channels of microwave echos: HH and HV. The values in the HH channel are the intensity of the horizontally echos of the horizontal transmitted microwave, whereas the HV channel is the intensity of the vertical echos of the same transmitted microwave.

A challenge of assessing the generalisation performance, given a dataset sampled from a single distribution, is that it does not reflect the case where the distribution of data under the expected testing conditions differs from the distribution of data sampled for training. Therefore, we split the dataset into three groups of discriminable classes, from which the images are sampled at different ratios between training and testing. We initially combine the two channels into one channel:

$$I(x, y) = \sqrt{I_{\text{HH}}(x, y)^2 + I_{\text{HV}}(x, y)^2} \quad (18)$$

where $I(x, y)$, $I_{\text{HH}}(x, y)$, and $I_{\text{HV}}(x, y)$ are the pixel values of the combined image, the HH image, and the HV image at (x, y) respectively. The dataset is then subdivided into three groups by hand for each class: (a) easily discriminable sets, (b) moderately discriminable sets, and (c) difficult cases (Figure 3).

Each of the groups is partitioned into training and testing splits and subsampled at different ratios, where specifically we distort the distribution of the training sets to simulate further

TABLE I: The number of samples in the experiment dataset separated by the test set and the three different training sets. The columns (a), (b), and (c) represent: easily identifiable samples, moderate samples, and difficult samples.

	Ship				Iceberg			
	(a)	(b)	(c)	total	(a)	(b)	(c)	total
Test	97	158	171	426	99	137	141	377
Train #1	96	15	17	128	99	13	14	126
Train #2	96	15	17	128	9	137	14	160
Train #3	96	15	17	128	9	13	140	162

Fig. 4: Visible images from [32] (domain transfer source).

imbalance and mismatch between the training distribution and the expected testing data distribution. These splits, and the corresponding skewed subsamplings, are shown in Table I.

In order to augment the training datasets using our proposed method, we use the satellite visible image dataset named DOTA [32], which is a collection of commercial satellite images containing many objects such as vehicles annotated with bounding boxes and class labels. Therefore we use visible and SAR image pairs with SAR images originating from the Statoil/C-CORE Iceberg Classifier Challenge dataset [31] and visible images from the DOTA [32] dataset. Due to the lack of iceberg visible images within either dataset, we pair iceberg SAR images from the Statoil/C-CORE Iceberg Classifier Challenge dataset [31] with representative non-ship images from the DOTA [32] dataset, for which purposes we use visible images of vehicles. Despite this obvious semantic mismatch in the second pairing, our I2I translation model specifically synthesises images conforming to the true distribution of the SAR iceberg images as enforced by the discriminator criteria of the loss function in Equation (11).

Initially, visible object images are extracted from the visible dataset using the annotations. Each extracted image is resized

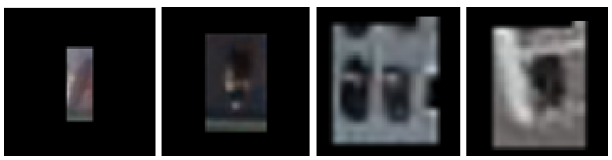


Fig. 5: Poor quality visible images illustrating blurriness and multiple objects (which we eliminate).

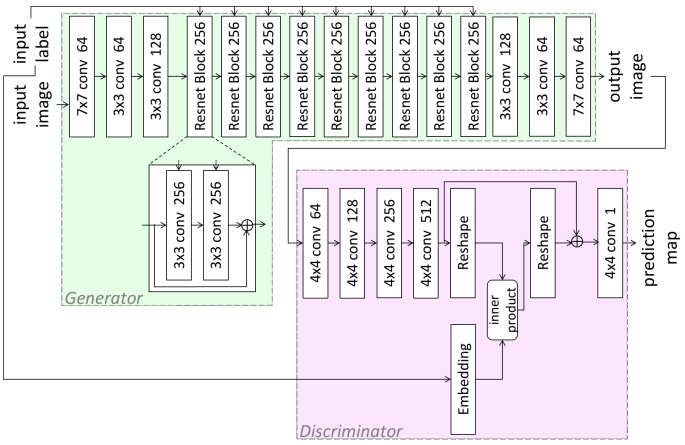


Fig. 6: Our network architecture:- Conditional Batch Normalisation layers are applied to every convolutional layer within the Generator whilst Instance Normalisation layers and Spectral Normalization are applied to every convolutional layer within the Discriminator.

in the same way as the SAR image, and its rotations adjusted accordingly. The backgrounds are set to black, which prevents including surrounding objects, which would be undesirable (Figure 4). The source domain visible dataset exhibits several images that are unclear or incorrect, as in Figure 5. Such images are eliminated based on their distances from the median of all of the images within each class. These distances are measured in the latent spaces trained by a Variational Autoencoder [33] on individual classes. Using the encoder, all of the images are embedded in a lower dimensional latent space that follows an approximate normal distribution, and the distances of each sample $d(x_i^c)$ are calculated:

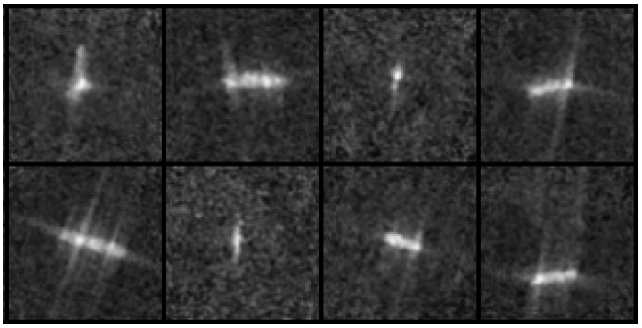
$$d(x_i^c) = \sqrt{(f_e^c(x_i^c) - \mathcal{M}^c)^T S^{c-1}(f_e(x_i^c) - \mathcal{M}^c)} \quad (19)$$

$$S^c = \mathbb{E}[(f_e(x_i^c) - \mathcal{M}^c)(f_e(x_i^c) - \mathcal{M}^c)^T] \quad (20)$$

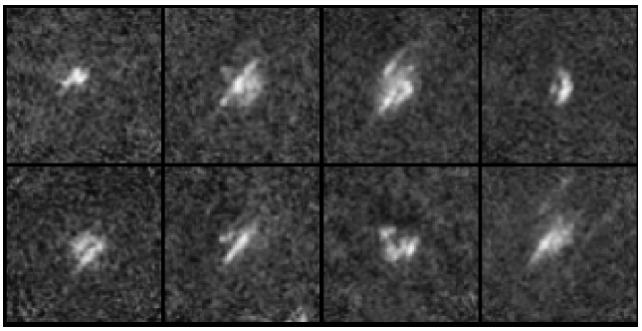
where x_i^c is the $i-th$ input sample of class c, f_e is the encoder, and $\mathcal{M}^c$ is the median of the encoded features in class c. S^c is a normalisation factor for each dimension of the feature vectors in class c. Half of the shorter distance samples are selected for each class, subsampling 14,034 visible ship images and 13,063 visible vehicles, resulting in clearer data and higher-quality annotations for use as our source domain.

B. Training Domain Transfer Model

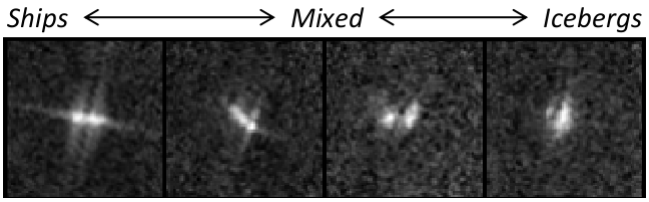
Domain transfer models, as described in Section III, are trained using the SAR images for each training split, where 1,500 ships and 1,500 vehicles images are subsampled from the visible images, prepared as previously outlined. The network architecture used in this experiment is shown in Figure 6, which follows a standard residual generative network, and the discriminator function uses Spectral Normalization on the convolutional layers. The network training parameters are: $\lambda_s = \lambda_t = 10.0$, $\lambda_{cyc} = 1.0$,



(a) Ships



(b) Icebergs



(c) Mixed

Fig. 7: Examples of the generated SAR images (Train #1): (a) and (b) are the individual class images. (c) are the inter-class images sorted by the class labels from ship to iceberg.

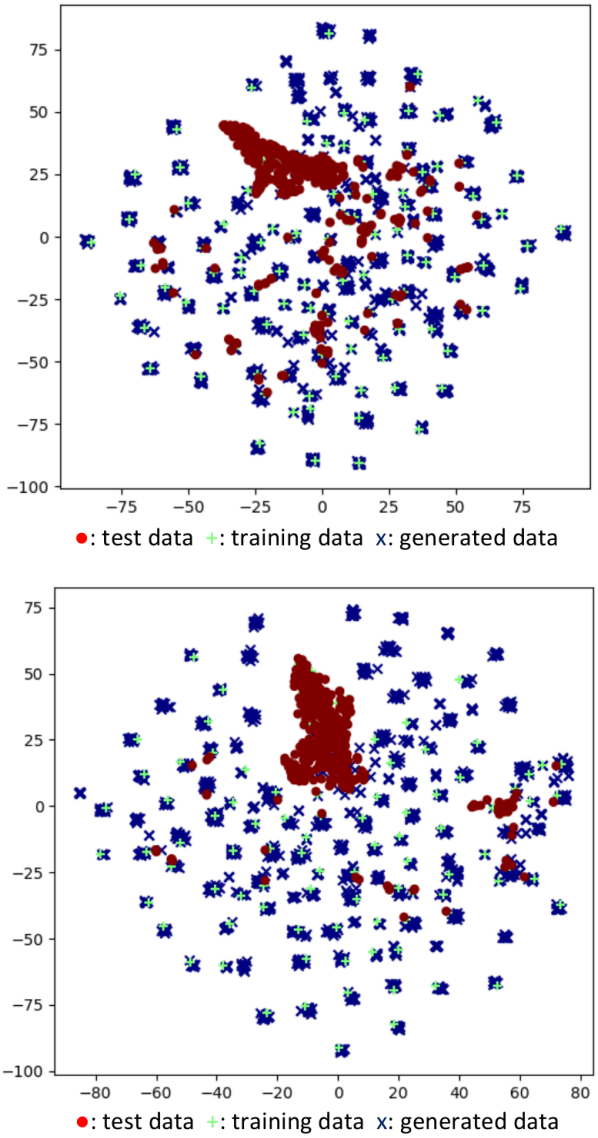


Fig. 8: t-SNE plot of ship (top) and iceberg (bottom) images from the test, training and generated datasets (Train #1).

$\lambda_{\text{gp}} = 0.01$, batch size $B = 32$, and number of critics $= 2$, 187,500 training iterations and optimised with Adam [34] (initial learning rate $\eta = 0.0001$, $\beta_1 = 0.5$, $\beta_2 = 0.999$).

C. Data Augmentation

Fake SAR images are synthesised using the visible images as the input of our transfer model, as discussed. This results in 3,000 generated SAR images, where examples of these generated images are shown in Figure 7. Additionally, we plot the real SAR images and fake SAR images using t-SNE [35] (Figure 8) to show how the different distributions interrelate. This plot shows that the fake SAR images are well-distributed around the real SAR images.

D. Evaluation on Object Classification Task

Evaluation of the classifier performance uses the simple Alexnet architecture [36], where the classifier performance is compared under the following conditions:

- BL: Only using the original training data [31]
- ROT: BL + rotated 90, 180, and 270 degrees
- MIXUP: Mixup [9] ($\alpha = 0.2$)
- MIXCG: BL + MixCycleGAN [27] (α=0.2)
- C2GMA: (Ours) BL + C2GMA (α=0.2, Section III)

The MixCycleGAN model in this experiment is trained with the same training parameters as our method uses.

The classifiers are trained with the three training datasets, as denoted in Table I, where the hyperparameters are optimised with the Stochastic Gradient Descent algorithm ($\eta = 0.02$, number of epochs $= 200$, $B = 512$). Performance is assessed via the testing dataset also outlined in Table I, using statistical accuracy (A), precision (P), recall (R) and F1-score (F1) (Table II).

TABLE II: Overall classification results: accuracy (A), precision (P), recall (R), and F1-score (F1) on the common test set for each of training sets #1–3.

	Train #1				Train #2				Train #3			
	A	P	R	F1	A	P	R	F1	A	P	R	F1
BL	0.715	0.746	0.725	0.735	0.469	0.469	0.500	0.484	0.469	0.469	0.500	0.484
ROT	0.707	0.723	0.714	0.719	0.469	0.469	0.500	0.484	0.469	0.469	0.500	0.484
MIXUP [9]	0.766	0.794	0.775	0.784	0.690	0.728	0.701	0.714	0.690	0.694	0.681	0.688
MIXCG [27]	0.760	0.765	0.764	0.765	0.757	0.783	0.766	0.776	0.676	0.708	0.687	0.697
C2GMA (Ours)	**0.800**	**0.807**	**0.804**	**0.806**	**0.771**	**0.795**	**0.779**	**0.787**	**0.691**	**0.729**	**0.703**	**0.716**

	Average			
	A	P	R	F1
BL	0.551 ± 0.142	0.562 ± 0.160	0.575 ± 0.130	0.568 ± 0.145
ROT	0.549 ± 0.137	0.554 ± 0.146	0.571 ± 0.124	0.562 ± 0.135
MIXUP [9]	0.715 ± 0.044	0.739 ± 0.051	0.719 ± 0.049	0.729 ± 0.050
MIXCG [27]	0.730 ± 0.048	0.752 ± 0.039	0.739 ± 0.045	0.745 ± 0.042
C2GMA (Ours)	**0.754 ± 0.056**	**0.777 ± 0.042**	**0.762 ± 0.053**	**0.769 ± 0.047**

Quantitative results are shown in Table II, alongside the additional individual per-class classification performances for ships and icebergs, shown in the confusion matrices in Figure 9. The overall results show that our proposed C2GMA data augmentation approach significantly outperforms the other approaches (BL, ROT, MIXUP [9], and MIXCG [27]). We find that generating new images using our approach increases training data appropriately, where the process of synthesising inter-class images is shown to provide significant improvements for the overall classification performance (C2GMA, Table II).

V. Conclusion

This paper proposes and evaluates a CycleGAN enabled data augmentation approach, Conditional CycleGAN Mixup Augmentation (C2GMA), to address the challenge of effective data augmentation within cross-domain imagery where the availability of one of the domains is limited. In particular, we show that the generation of interpolated mixed class (non-visible domain) image examples via our novel C2GMA methodology leads to a significant improvement in the quality of non-visible domain classification tasks that suffer due to limited data availability and variety. Focusing on classification within the synthetic aperture radar domain, our approach is evaluated on a variation of the Statoil/C-CORE Iceberg Classifier Challenge dataset and achieves 75.4% accuracy, demonstrating a significant improvement when compared against traditional augmentation strategies. Future work will consider DNN architecture modifications to enable generation of higher quality images for improved classification results and applications to other non-visible band imaging domains.

References

[1] R. G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 18, no. 7, pp. 690–706, 1996.

[2] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 3, pp. 334–352, 2004.

[3] A. Vibhute and S. K. Bodhe, "Applications of image processing in agriculture: a survey," *Intl. Journal of Computer Applications*, vol. 52, no. 2, 2012.

[4] S.-H. Huang and Y.-C. Pan, "Automated visual inspection in the semiconductor industry: A survey," *Computers in industry*, vol. 66, pp. 1–10, 2015.

[5] I. G. Goodfellow, Y. Bengio, and A. C. Courville, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[6] M. E. Kundegorski, S. Akçay, G. P. de La Garanderie, and T. P. Breckon, "Real-time classification of vehicles by type within infrared imagery," in *Proc. Optics and Photonics for Counterterrorism, Crime Fighting, and Defence XII*, 2016.

[7] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, "Target classification using the deep convolutional networks for sar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4806–4817, 2016.

[8] S. Akçay, M. E. Kundegorski, C. G. Willcocks, and T. P. Breckon, "Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2203–2215, Sep. 2018.

[9] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. 6th Intl. Conf. on Learning Representations*, 2018.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, 2014.

[11] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Intl. Conf. on Computer Vision*, 2017.

[12] J. Shijie, W. Ping, J. Peiyi, and H. Siping, "Research on data augmentation for image classification based on convolution neural networks," in *Proc. IEEE Chinese automation congress*, 2017.

[13] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *CoRR abs/1708.04896*, 2017.

[14] T. Devries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *CoRR abs/1708.04552*, 2017.

[15] P. Chen, S. Liu, H. Zhao, and J. Jia, "Gridmask data augmentation," *arXiv preprint arXiv:2001.04086*, 2020.

[16] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Dropblock: A regularization method for convolutional networks," in *Advances in Neural Information Processing Systems 31*, 2018.

[17] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE Intl. Conf. on Computer Vision*, 2019.

[18] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *Proc. 36th Intl. Conf. on Machine Learning*, 2019.

[19] P. T. G. Jackson, A. A. Abarghouei, S. Bonner, T. P. Breckon, and B. Obara, "Style augmentation: Data augmentation via style randomization," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, 2019.

[20] G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, and J. Shlens, "Exploring the structure of a real-time, arbitrary neural artistic stylization network," in *Proc. British Machine Vision Conf.*, 2017.

[21] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation

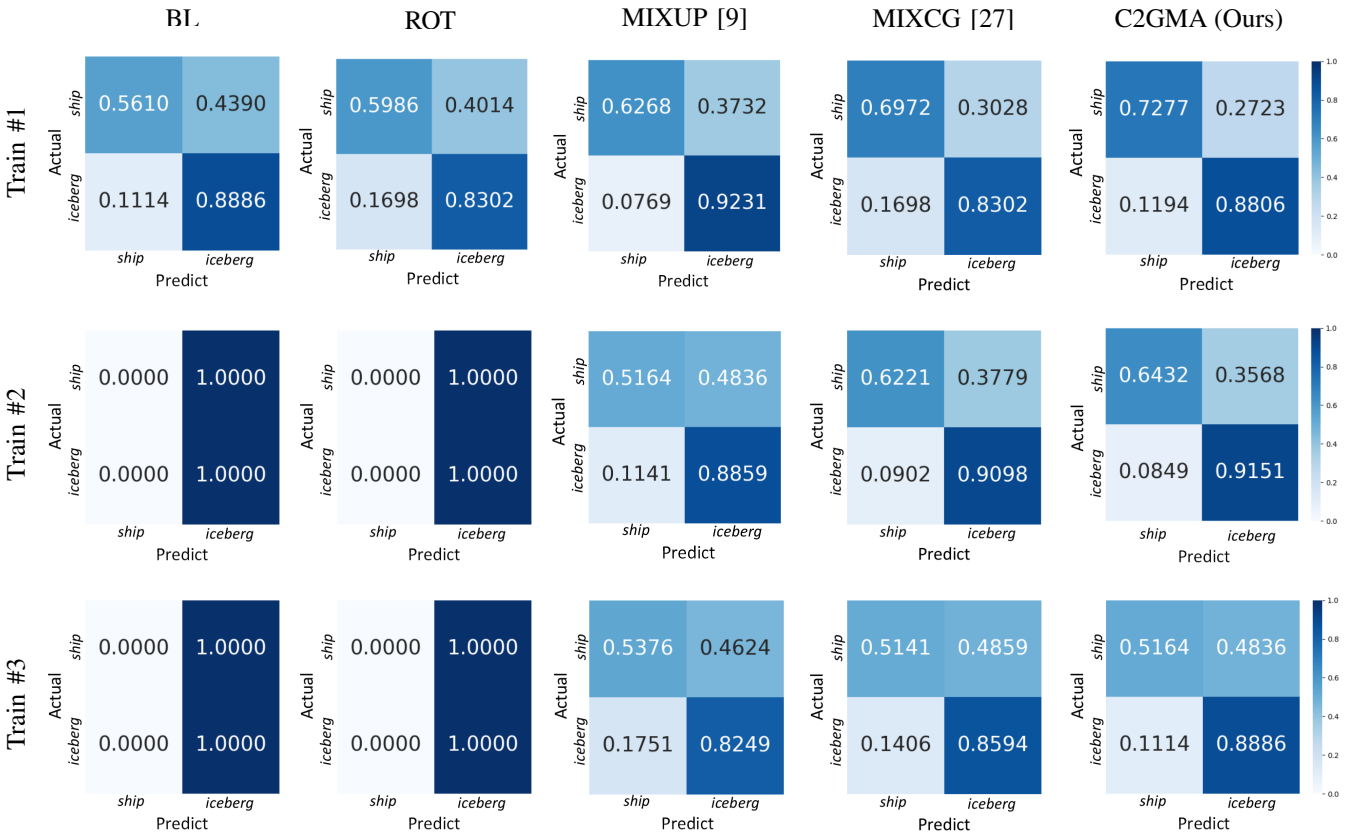


Fig. 9: Per-class performance (confusion matrices) of our approach (C2GMA) against prior work in the field.

learning with deep convolutional generative adversarial networks," in *Proc. 4th Intl. Conf. on Learning Representations*, 2016.

[22] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR abs/1411.1784*, 2014.

[23] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Intl. Conf. on Machine Learning*, 2017.

[24] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *Proc. 5th Intl. Conf. on Learning Representations*, 2017.

[25] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2019.

[26] T. Miyato and M. Koyama, "cGANs with projection discriminator," in *Proc. 6th Intl. Conf. on Learning Representations*, 2018.

[27] D. Liang, F. Yang, T. Zhang, and P. Yang, "Understanding mixup training methods," *IEEE Access*, vol. 6, pp. 58 774–58 783, 2018.

[28] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *Proc. 6th Intl. Conf. on Learning Representations*, 2018.

[29] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems 30*, 2017.

[30] C. Chu, K. Minami, and K. Fukumizu, "Smoothness and stability in gans," in *Proc. 8th Intl. Conf. on Learning Representations*, 2020.

[31] Statoil/C-CORE. Statoil/c-core iceberg classifier challenge. [Online]. Available: https://www.kaggle.com/c/statoil-iceberg-classifier-challenge.

[32] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "Dota: A large-scale dataset for object detection in aerial images," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2018.

[33] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, "Variational autoencoder for deep learning of images, labels and captions," in *Advances in Neural Information Processing Systems 29*, 2016.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Intl. Conf. on Learning Representations*, 2015.

[35] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012.