3-1-2023

Texture-based latent space disentanglement for enhancement of a training dataset for ANN-based classification of fruit and vegetables

Khurram Hameed
Edith Cowan University, k.hameed@ecu.edu.au

Douglas Chai
Edith Cowan University, k.hameed@ecu.edu.au

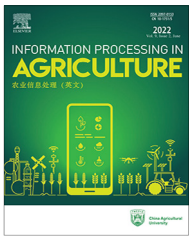
Alexander Rassau
Edith Cowan University, a.rassau@ecu.edu.au

10.1016/j.inpa.2021.09.003
Hameed, K., Chai, D., & Rassau, A. (2023). Texture-based latent space disentanglement for enhancement of a training dataset for ANN-based classification of fruit and vegetables. Information Processing in Agriculture, 10(1), 85-105. https://doi.org/10.1016/j.inpa.2021.09.003

Available at www.sciencedirect.com

INFORMATION PROCESSING IN AGRICULTURE 10 (2023) 85–105

journal homepage: www.elsevier.com/locate/inpa



Texture-based latent space disentanglement for enhancement of a training dataset for ANN-based classification of fruit and vegetables

Khurram Hameed *, *Douglas Chai*, *Alexander Rassau*

School of Engineering, Edith Cowan University, 270 Joondalup Drive, Joondalup WA 6027, Perth, Australia

ARTICLE INFO

Article history:
Received 3 August 2020
Received in revised form
13 September 2021
Accepted 28 September 2021
Available online 5 October 2021

Keyword:
Information Maximisation (IM)
Fruit and vegetables classification
Representation Learning (RL)
Variational Autoencoder (VAE)
Generative Adversarial Network (GAN)
Latent space disentanglement

ABSTRACT

The capability of Convolutional Neural Networks (CNNs) for sparse representation has significant application to complex tasks like Representation Learning (RL). However, labelled datasets of sufficient size for learning this representation are not easily obtainable. The unsupervised learning capability of Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) provide a promising solution to this issue through their capacity to learn representations for novel data samples and classification tasks. In this research, a texture-based latent space disentanglement technique is proposed to enhance learning of representations for novel data samples. A comparison is performed among different VAEs and GANs with the proposed approach for synthesis of new data samples. Two different VAE architectures are considered, a single layer dense VAE and a convolution based VAE, to compare the effectiveness of different architectures for learning of the representations. The GANs are selected based on the distance metric for disjoint distribution divergence estimation of complex representation learning tasks. The proposed texture-based disentanglement has been shown to provide a significant improvement for disentangling the process of representation learning by conditioning the random noise and synthesising texture rich images of fruit and vegetables.

1. Introduction

Computer vision based classification of fruit and vegetables has significant applications in the food industry for numerous applications including quality assessment, fruit and vegetables grading, robotic harvesting and self-checkouts at supermarkets. The data used for these applications range from binary to hyperspectral images. Machine learning techniques have been exploited for both Deep Convolutional Neural Networks (DCNNs) [1,2] and statistical classification techniques [3,4]. Detailed surveys on computer vision based food industry applications can be found in [5–16]. The classification of fruit and vegetables has notable inherent challenges as compared to other computer vision tasks due to large numbers of potential variations in texture shape and colour [17,18,87]. These challenges result in a large number of features being required to differentiate the classes. These features must come from the training samples, and hence require large training datasets.

* Corresponding author.
E-mail addresses: k.hameed@ecu.edu.au (K. Hameed), d.chai@ecu.edu.au (D. Chai), a.rassau@ecu.edu.au (A. Rassau).
Peer review under responsibility of China Agricultural University.
https://doi.org/10.1016/j.inpa.2021.09.003

Carefully estimated handcrafted features have been used with a variety of classification algorithms and have shown state-of-the-art results [19,20]. However, with the increase in application domains of machine learning, the probability of success of handcrafted feature methods reduces due to their inherent limitations of time complexity, brittleness and incompleteness. A significant result has been presented in [21], where a joint distribution across n variables represented by k variables shows a uniform nature however, variations can be discovered if more data is considered, i.e. handcrafted features are insufficient and can be incomplete for a large number of classes. The need for more training samples can also be explained by the curse of dimensionality principle in machine learning problems, where the required number of samples increases exponentially with the number of features to be extracted, whereas the performance increases logarithmically [22]. As concluded in [23], a learning model that depends upon local generalisation with a maximally varying feature vector of n entries requires (2^n) training samples. The mathematical explorations of a Gaussian kernel based learning algorithm w.r.t. a smoothness prior was performed for a function with $2k$ zero-crossing along a straight line. Smoothness prior is among the most popular assumptions in low level vision applications measured as the energy term for the number of times the assumption is violated. This term is then used as an estimator to find the lower bound on the number of training samples required.

Convolutional Neural Networks (CNNs) have an inherent ability to deal with large application domains and number of features due to their latent distributed representation capability. This distributed representation is performed by n-k active neurons out of n neurons which is a form of sparse representation. Hence, with the distributed representation capability of CNNs, the representation capability grows exponentially w.r.t. the number of neurons. Based on the distribution capability of a neuron they can also generalise to a region with no prior information. CNNs have a capability of learning the features at lower levels to perform the representation and this capability grows with the depth of the network [24]. Other significant motivations to use CNNs are their capability to estimate invariant representation and sparsity to cure the curse of dimensionality, which eventually is required to meet the goal of better representation. Based on this basic principle, posterior probabilities learned up to the penultimate layer are used for decision making in the last layer. The learned posterior probabilities can be considered as a manifold for continuous data, i.e. signals or concentrated regions for discrete data, e.g. text processing. However, availability of large quantities of labelled data is the crucial limitation for training supervised learning CNNs to estimate the manifold.

Techniques like data augmentation [25] and transfer learning [26] have been used for data enhancement at initial levels to overcome the scarcity of labelled datasets. However, the augmentation and transfer learning approaches are not always meaningful for some applications, e.g. fruit and vegetables classification considering the case where classification is based on colour and texture information only. More recent techniques try to formulate the complex data distributions as functions of noise sampled from a Gaussian or Normal probability distribution. Variational Auto Encoders (VAEs) [27] and Generative Adversarial Networks (GANs) [28] are the most recent examples of these techniques. Both techniques learn reusable feature representations while being unsupervised in nature, hence no large labelled dataset is required. These learned feature representations can be used in many auxiliary applications such as semi-supervised learning, in painting and to generate unseen data samples for supervised learning tasks, e.g. vision based classification of fruit and vegetables. However, VAEs have an inherent limitation of assigning large probability to non-data points in a distribution, hence causing atypical blur in images during the synthesis process [29]. A method to address this limitation of VAEs has been suggested by the GAN design in [28] through application of the Jensen-Shannon Divergence (JSD), where small probabilities are also assigned to data samples. Recently, a detailed study has been presented on low magnitude GAN perturbations to generate visually salient image features in [30]. The discriminative features learned from the natural images are added to the GAN perturbations as a global feature in the first step. These learned perturbations are then refined for complex image regions to generate images with greater details in the second step. The state-of the-art robust DCNN classifiers including VGG-16 and ResNet50 have been tested for the proposed attack-based perturbations accumulation technique. Note that VGG-16 is a 16-layer deep CNN architecture named after the Visual Geometry Group while ResNet50 is a 50-layer deep variant of the Residual Networks. The appropriate feature selection is an important concept and has significant applications. Feature selection techniques are presented in [31] for network intrusion detection. A Sparse Logistic Regression (SPLR) based feature selection technique has been proposed on a subset based network intrusion feature extractor and selector. These features are then used for final discriminative feature selection for improved performance. An improved approach based on a similar concept has been presented in [32]. The attack structures are used with SPLR to improve the feature selection for network intrusion detection. The proposed technique is then used for Support Vector Machine (SVM) based classification to detect intrusion. A similar application of the appropriate feature selection is explained in [33].

In this research, we investigate the reuse of a manifold learned by unsupervised VAEs and GANs. The learned manifold is used to generate Representation Learning (RL) for data enhancement, and later can be extended for feature extraction to classify fruit and vegetables, e.g., at a supermarket self-checkout as a potential application. The concept of latent space disentanglement is combined with the style transfer, considering the strengths of the batch normalisation in controlling the behaviour of the latent space GANs generation process. The concept of batch normalisation, Information Maximisation (IM) and style transfer have been studied in detail separately, however, their combination still needs to be explored in more detail. Various architectural variants of VAEs and GANs are also investigated for learning the representations. The classification of fruit and vegetables at supermarket self-checkouts has presented significant challenges due to two basic limitations: (a) the fruit and vegetables in

supermarkets are sold in a huge number of classes and varieties and have highly inconsistent features within each class like shape, colour and texture, and (b) factors like ambient light and the scanning process are also highly inconsistent at self-checkouts. Additionally, the labelled datasets of fruit and vegetables for supermarket self-checkouts are scarce and building a dataset is a time consuming and expensive task. The application of GANs and VAEs to enhance datasets and synthesise the samples is a novel approach to enhance this task. The extended applications of statistical distribution learning can be data augmentation [34] and feature representation for classification.

The rest of the paper is organised as follows: the detailed description of the significant efforts of RL is provided in Section 2. The details on the dataset used and basic preprocessing relevant to the application is presented in Section 3.1. A discussion on the state-of-the-art techniques implemented for the experiments is given in Sections 3.2 and 3.3. The proposed approach of texture-based disentanglement is explained in Section 3.4. A comparison of results based on our dataset is presented in Section 4. The conclusion based on the implementation and results including the future directions is provided in Section 5.

2. Related work

Unsupervised representation learning and its applications are essential parts of machine learning and computer vision. Graphical and energy based models are among the first ideas for generative models for representation learning based on latent space for example the Restricted Boltzmann Machine (RBM) [35] and Deep Belief Network (DBN) [36]. In these models, expensive Markov Chain Monte Carlo (MCMC) techniques have been used to deal with the intractable gradients in order to extract inference and normalisation constants. The DCNNs have a capability to be implemented as parameterised functions to generate the samples and overcome the issues of RBM and DBN. Probabilistic graphical models are used in VAEs where the lower bound of data likelihood is maximised to achieve the goal [27,37]. Recently a better RL technique has been reported for Pixel Recurrent Neural Network (PRNN) based on an autoregressive model [38]. VAEs and GANs are among a few prominent approaches that have emerged recently. Other significant approaches include Generative Stochastic Networks (GSNs) [39], PRNNs [38] and Pixel Convolutional Neural Networks (PCNNs) [40]. However, training instability of GANs limits the generation of high resolution representations, and much research has been reported to improve the training process in GANs [41–44].

Conditional generative models have also shown significant improvement for reconstruction of images from noise based on the conditioning variables, i.e. class labels or visual attributes [38,45,46]. Other significant work uses the image as a condition to generate images for different applications like photo editing [47], style transferring [48] and super resolution [49]. However, all of these techniques are limited to low resolution and size for the feature vector learned. A more recent approach has used a stacked GAN as a series of two stages to generate more detailed feature vectors [50]. The random normal distribution is conditioned with unstructured text to generate the representation. As proposed, the first stage generates the primitive shape and colour information for images and the second stage improves the low resolutions regions. This method has a significantly higher resolution of 256×256 pixels as compared to other methods based on text conditioning [51,52].

A series of GANs is also used to generate images. In this approach the image generation task has been separated into structure and style generation [53]. A Laplacian pyramid based series of GANs has been proposed where the output image of the former stage is conditioned and added to the input image as an input of the current stage [54]. A multilevel pre-trained discriminator network has been used on different spatial resolutions to generate images of 32×32 pixels from a series of GANs. A progressively growing GAN has been proposed that starts with a low resolution image of 4×4 pixels and progressively adds layers to the network. The network is programmed to shift attention to the new finer details with the addition of each new layer to ultimately generate images of $1\,024 \times 1\,024$ pixels [55]. These GANs have also set a benchmark performance with semi-supervised learning and image and visual description based joint modelling [44].

The IM has been used recently in [46], where the IM is used to improve the correspondence between semantic features of data and the individual dimension of latent z. The latent code c is concatenated with incompressible noise z to constrain the use of z for image synthesis. The latent code is used to factorise the noise z in a disentangled way. The overall process is completely unsupervised; a similar approach is also presented in [56]. Considering the similar concept of learning disentangled latent code the most recent approach has been reported in [57]. A non-linear mapping based disentangled latent space $\mathscr{W}$ has been generated by inputting a continuous non-compressible noise signal. The non-linear mapping technique used in [57] is important to achieve good results, however, dealing with the complex datasets of fruit and vegetables requires significant texture details. Specifically, in our application of GANs for fruit and vegetable images synthesis, texture details are very important due to similar texture features for multiple different fruit and vegetables. Considering the significance of texture details in the synthesised images, we have proposed a texture based non-linear mapping of latent space. Previously mentioned state-of-the-art techniques [46,56,57] have been proposed for non-linear mapping of the Gaussian based latent codes where texture based latent space disentanglement needs a detailed exploration. In this research, we have used texture details of fruit and vegetable images as latent code to disentangle the latent space for synthesis of texture rich fruit and vegetable images.

3. Proposed method

A discussion on the dataset used for experiments and testing along with the proposed texture-based latent space disentanglement is described in this section. The latent space disentanglement process is discussed in detail considering the state-of-the-art VAE and GAN techniques.

3.1. Dataset and Pre-processing

The experiments have been performed on a 4 000 images dataset of 20 different fruit and vegetable classes (200 images per class) including Apple, Broccoli, Brown Onion, Capsicum, Cauliflower, Eggplant, Hass Avocado, Honey- dew Melon, Iceberg Lettuce, Kent Pumpkin, Kiwifruit, Lemon, Orange, Packham Pear, Potato, Rockmelon, Tomato, Topless Pineapple, Watermelon and Yellow Banana with a base resolution of 640 × 640 pixels. The images have been captured with a mobile phone camera (Huawei P9 lite) in the local supermarket (Joondalup, Australia) to capture a real-life supermarket environment. To maintain reasonable environmental and lighting consistency for the dataset acquisition process, all images have been captured during 10:00 am to 03:00 pm using the ambient supermarket lighting. An approximate distance of 35 cm between the fruit surface and mobile capturing device has been maintained where the fruit and vegetables were captured in their respective shelves. A brief description of the image capturing device and environmental conditions for dataset acquisition is presented in Table 1. The obtained images of resolution 3 120 × 4 160 pixels are manually inspected and segmented to 640 × 640 pixels (base resolution) to remove any background and occlusion at the first stage (see Fig. 1). The segmented images are further sliced to a resolution of 160 × 160 pixels this assumption is made to overcome the significant size variation for fruit and vegetables, illustrated in Fig. 2. These sliced images are then used for training and groundtruth, where an example sliced image of each class is shown in Fig. 3. This two-stage image acquisition and segmentation process results in 64 000 images (3 200 images per class) that after the slicing consist of colour and texture details only. Smaller resolution patches have significant benefits when dealing with inconsistent imaging, surface defects or damage, texture, colour and size variations, illumination changes and occlusions in the supermarket environment. An example of variations of associated physical aspects is that an image of 640 × 640 pixels cannot be used to capture the shape of both a watermelon and grapes when the images are captured at similar distance due to the significant size differences of the two fruits. Considering the challenges associated with the environment, multiple image patches of a fruit or vegetable can be selected to verify and cross check the results. The smaller image size will also help in model size reduction and allow faster training and synthesis processes. The approach is further based on the concepts of data efficient Artificial Neural Network (ANN) training and Constructive Predictive Coding (CPC) [58,59]. The representation of high dimensional data, i.e. high resolution images to a latent vector of much lower dimension, causes the loss of significant information. The concept of slicing and generating an intermediate level representation preserves the significant temporal and spatial information. A similar concept has been provided in [60] for grey-scale to RGB image transformation.

3.2. Variational Autoencoders (VAEs)

Autoencoders (AEs) are unsupervised learning models in their simplest form with only one linear transformation layer as a single hidden layer along with an input and output layer. Considering the above definition as the simplest representation of an AE, this can be described as:

$$\mathrm{AE}\left(\theta,\theta^{'}\right)=\frac{1}{\mathrm{N}}\sum_{n=1}^{\mathrm{N}}||I_n-\mathrm{D}_{\theta^{'}}\left(\mathrm{E}_{\theta}(I_n)\right)||^2, \tag{1}$$

where E_{θ} and $\mathrm{D}_{\theta^{'}}$ are the encoder and decoder function, respectively, which are simple linear transformation functions, i.e. $\mathrm{ReLU}(\mathrm{W}I_n+\mathrm{b})$ with weights W and bias b. The learned parameters of the encoder and

decoder are denoted by θ and $\theta^{'}$ for a dataset of N samples, i.e., 64 000 in our case. The goal here is to convert the coded feature vector back to the same feature vector and so the difference between the original features I_n and the generated features is minimised to achieve this goal. However, AEs have a rigid behaviour towards representation of a disjoint distribution from the training dataset due to their limitation in detecting a valid latent vector. VAEs can be used to learn the representation from a latent vector, which can be used to generate new samples with disjoint latent space for data enhancement. The encoder and decoder in a VAE are converted to probabilistic estimators for better RL. The VAE with probabilistic encoders and decoders can be described as:

$$\mathrm{VAE}(\theta,\phi)=-\mathbb{E}_{z\sim q_{\phi}(z|I_n)}\left[\mathrm{p}_{\theta}(I_n|z)+\mathrm{D}_{\mathrm{KL}}\left(\mathrm{q}_{\phi}(z|I_n)||p_{\theta}(z)\right)\right], \tag{2}$$

where p_{θ} and q_{ϕ} are the probabilistic encoder and decoder, respectively. The input feature vector is denoted by $I_n$which in our case is simply a vector of three channel values for the RGB image, i.e. the size of this vector is 160 × 160 × 3. The learned parameters of the encoder and decoder are represented by θ and ϕ, respectively where Kullback–Leibler divergence is denoted by D_{KL}. A latent vector z is used in proposed approach for RL described in (6). The basic architectures of AEs and VAEs are described in Fig. 4. This architecture of a VAE can easily overcome the main issue of AEs by considering that the new disjoint latent vector is sampled from a known probability distribution. This is achieved by simply represent-

Table 1 – Description of mobile image capturing device and environmental conditions.

Image sensor/condition	Description
Mobile device	Huawei P9 lite
Vision sensor	Sony IMX214 Exmor RS
Image resolution	3 120 × 4 160, 640 × 640 and 160 × 160 pixels
Device distance	35 cm from fruit/vegetable surface
Lighting condition	Supermarket ambient lighting (10:00 am to 03:00 pm)
Location	Local supermarket (Joondalup, Australia)

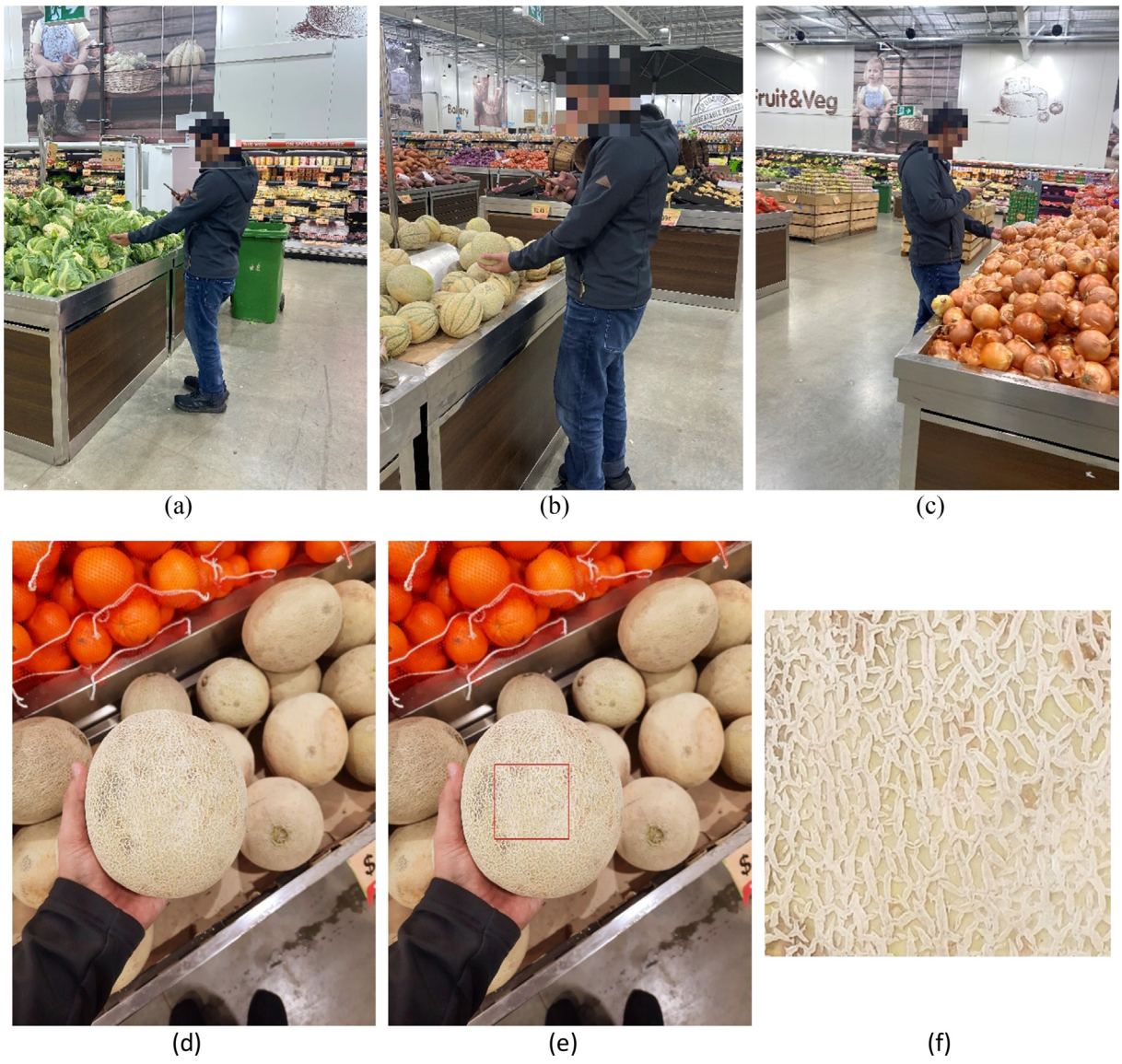


Fig. 1 – Image data acquisition in a real-life supermarket environment: (a), (b) and (c) image acquisition at supermarket fruit and vegetable shelves, (d) captured high resolution (3 120 × 4 160 pixels) fruit image, (e) Region of Interest (ROI) representation for manual cropping, and (f) cropped image (640 × 640 pixels).

ing each feature of a training dataset by a probability distribution. A dense VAE is implemented for our experiments with an input feature vector of size 160 × 160 × 3 that passes through a linear transformation layer, i.e. a dense layer to estimate the mean (μ) and standard deviation (σ) as statistical features with a dimension of 2 for both. The behaviour of a single dense layer VAE is studied for 256, 512, and 1 024 nodes.

$(q_\theta(z|I_n))$ and decoder $(P_\theta(I_n|z))$

The spatial details in an image are vital, especially for classification from colour and texture only. A 16-layer convolutional VAE based on [61] is implemented for RL. The convolutional VAE is implemented as a set of convolutional and transposed convolutional layers for the encoder and decoder, respectively. The detailed architecture along with filter sizes of the implemented convolutional VAE is described in Table 2. Four convolutional layers extract the abstract lower level features of an image before estimating the statistical details for RL. The decoder is an inverse of the encoder network and transposed convolutional [62] is used to up scale the feature vector. Based on the approach of [61], the encoder in the VAE can be described as a set of convolutional, batch normalisation and activation operations as follows:

$$R_l^n = I_n * F_l, n = 1, 2, 3, \cdots, N \text{ and } l = 1, 2, 3, \cdots, L, \tag{3}$$

$$\mu_{R_l^n} = \frac{1}{m}\sum_{1}^{m} R_m^l, \tag{4}$$

$$\bar{R}_n^l = \mathrm{ReLU}\left(\mu_{R_l^n}\right), \tag{5}$$

where R_l^n is the representation learned by convolution layer l with filter F_l from L layers on an image I_n from the dataset of N

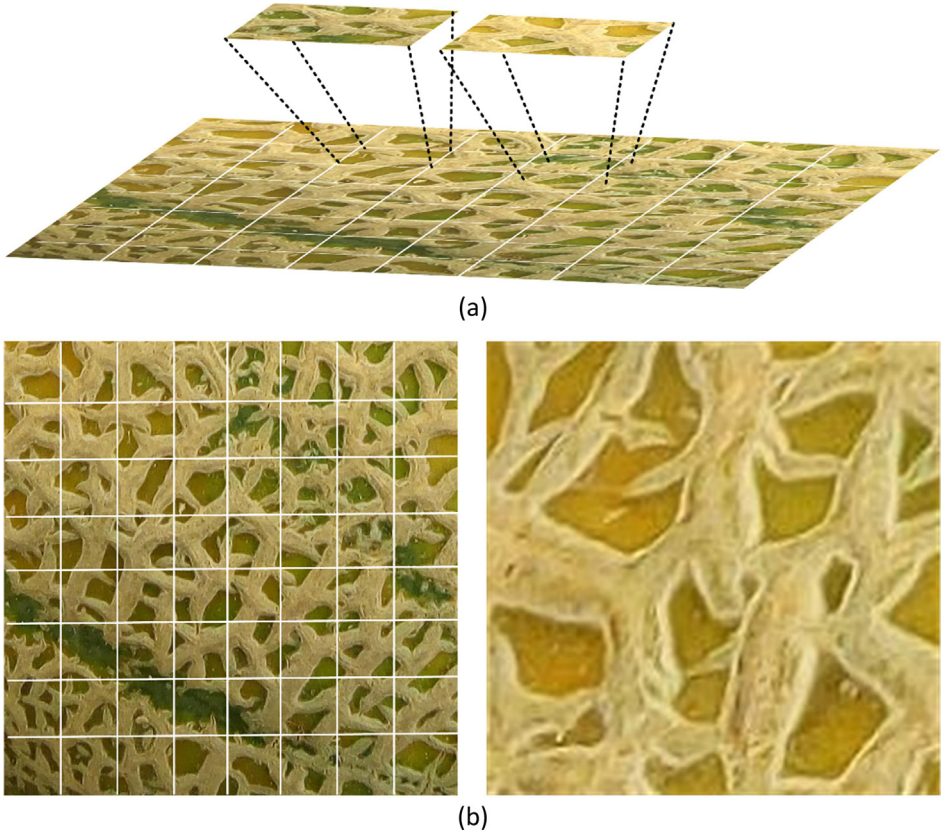


Fig. 2 – Visualisation of image slicing of a Rockmelon: (a) from a 640 × 640 pixels image to 16 patches of 160 × 160 pixels, and (b) a comparison of sliced and groundtruth images.

images. The leaned representation is then normalised for m elements in R_l^n to estimate the final representation $\bar{R}_n^l$ based on the ReLU activation function. The sampling from normal distribution $\mathcal{N}$ for estimation of statistical features to define the latent vector z in the designed network can be denoted as:

$$z \sim \mathcal{N}\left(\mu_\theta\left(\bar{R}_n^l\right), \phi_\theta\left(\bar{R}_n^l\right)\right), \tag{6}$$

where mean and standard deviation of R_l^n based on θ learning parameters are represented as μ_θ and ϕ_θ, respectively. The decoder is then used to convert a learned latent vector z back to the input feature vector. The decoder network is based on transposed convolution upsampling and can be described as:

$$f_v = \sum_{l=1}^{L} z \oslash a_l, \tag{7}$$

where $\oslash$ represents the transposed convolution between a latent vector z and the activation filter a_l. This transposed convolution has been repeated four times to convert the latent vector to an input feature vector, i.e. images in our case. The transposed convolutions are followed by batch normalisation and activation functions as defined in Eq. (4) and (5).

3.3. *Generative Adversarial Networks (GANs)*

The AE and VAE based models generate an input feature vector from a dataset space S by defining a low dimensional representation z for each of the images I_n in the space. The complex marginal distribution $p(\in)$ overall latent vectors Z are ideally supposed be known. So, the new feature vectors f_v are generated by sampling $p(\in)$ and feeding this into the decoder. However, there are significant limitations of AEs and VAEs in terms of requirement for: (a) a precisely trained decoder that can generate realistic feature vectors, and (b) a match between $P(Z)$and $p(\in)$, i.e. these distributions should overlap. Significant efforts have been reported to meet both challenges by minimising the AE and VAE loss and by either modifying the encoder or reducing the difference between the two distributions [63]. More rigorous penalties have been applied on mismatched distributions by converting Kullback–Leibler (KL) divergence to Evidence Lower Bound Objective (ELBO) in [64].

GANs in comparison generate the input feature vector from high dimensional data distribution with a typical architecture of two networks: (a) generator and (b) discriminator. A latent code based input feature vector is generated by a generator with an indistinguishable distribution from the training data distribution. A discriminator network is used to discriminate between fake and original feature vectors, where gradient information during the discrimination process can be used for fine tuning both networks for effective results. The generator is the most significant part of this network however, the adaptive loss function learned by the discriminator can be neglected once the training process is completed. A basic architecture of a GAN is shown in Fig. 5.

Fig. 3 – Example sliced images (160 × 160 pixels) of 20 classes of fruit and vegetables.

The vital element to reconstruct the input feature vector is the distance reduction between the distribution of the original and fake data. Multiple formulations have been derived for estimation of the distance between these distributions. However, the task is non-trivial if both distributions are not easily distinguishable or do not have significant overlapping. Jensen-Shannon (JS) divergence was originally used [28] as a distance metric between the distributions. More stable distribution divergence techniques include least squared [65], margin absolute deviations [66] and Wasserstein distance [67]. Architectural variations have also been tested for high resolution and improved feature representations [50,55]. However, higher resolution feature vectors (i.e. images) are easily distinguished form the originals by a discriminator network. The convolution used in the GANs also limits the process in terms of model size, training time and variations over resolution [68]. We have applied the state-of-the-art techniques for input feature vector regeneration for a fruit and vegetables dataset. Starting from the initial effort in [28] we have compared Wassertein Generative Adversarial Network (WGAN) [67] and Information Maximisation Generative Adversarial Network (InfoGAN) [46,69] as two significant variants of GAN.

Considering the problem of unsupervised learning the set of some parametric probability densities are estimated to achieve the maximum likelihood with examples of ground-truth data [67]. In our case the above statement can be described as:

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log P_{\theta}(I_n), \tag{8}$$

where P_θ is the set of parametric probability distributions and I_n is an example of original data drawn from a dataset of N elements. GANs also used a similar techniques for feature vector generation by considering P_{In} as the real data distribution and P_θ as the parametric distribution and applying a distance metric to reduce the distance between the two overlapping probability distributions as the loss, e.g. KL divergence for VAEs and JS divergence for GANs. However, there can be many real-life examples where the distributions of a real input feature vector and the generated feature vector are disjoint or not overlapping. This issue is resolved by many state-of-the-art GANs by adding noise distributions however this can affect the quality of the generated feature vector [67]. Moreover, if the distributions are disjoint, no parameter updating takes place in the initial stages of the training process [70]. Considering the loss function of a GAN illustrated in Fig. 5, the above statement can be described as:

$$\text{loss}(G, D) = \mathbb{E}[\log(I_n)] + \mathbb{E}[\log(1 - D(G(z)))], \tag{9}$$

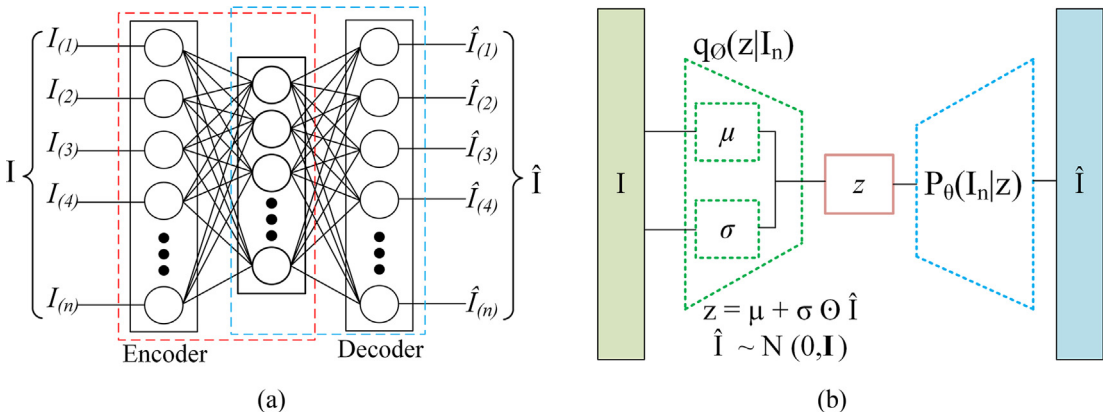


Fig. 4 – Basic architectural description of: (a) single layer autoencoder (AE) for encoding and decoding feature vector (I) to (Î), respectively, and (b) variational autoencoder (VAE) with probabilistic encoder.

where the discriminator for an ideal GAN (D_ϕ) can be described as:

$$D_\phi = \left[\frac{P_{I_n}}{P_{I_n} + P_{\hat{I}_n}}\right], \tag{10}$$

where $P_{\hat{I}_n}$ is the probability distribution of the generated feature vector. Substituting (10) into (9), it can be evaluated that the loss term is zero for both KL and JS divergence as distance metrics, i.e no parameter updating for discriminator or generator. The Wasserstein distance, however is a more stable distance metric if the distributions are disjoint, measuring the horizontal distance between the means of both distributions. This horizontal distance can be used as a measure of the difference between distributions for GANs. Based on the above discussion it can be concluded that the Wasserstein distance metric is more suitable for better and faster convergence of both discriminator and generator meaning it is more suitable for complex problems like representation learning for fruit and vegetables. Hence, a Wasserstein GAN is implemented in this study for representation learning of a fruit and vegetables dataset. The architectural details of the implemented Wasserstein GAN are described in Table 3.

The process of RL has been studied as a mutual information inclusion for latent vector generation in [46]. An assumption has been made that the generator can use the noise vector z as an input in a highly entangled manner for generation of the representation. To improve disentanglement, statistical semantic information s has been used as a condition on the noise z to limit the randomness in the distribution. This process provides significant benefits for RL of complex tasks with a high number of classes through production of effective feature vectors that can be used as novel samples. Moreover, the samples among the multiple classes have some inherent meaningful statistical factors and have distinct physical characteristics. The colour of fruit and vegetables for example have some specific combinations of probability distribution for the three channels in RGB images, hence additional statistical factors (e.g. shape or size) can improve the effectiveness. Instead of using the random noise vector, the InfoGAN uses random noise z and statistical semantics of data distribution as a condition on processing the noise vector. Considering the random noise as z and set of statistical semantic features as s, the objective function for InfoGAN (V_I) [46] can be described as:

$$\min_G \max_D V_I(D, G) = V(D, G) - \lambda I(s; G(z, s)), \tag{11}$$

where G and D represent the generator and discriminator, respectively, and I is the mutual information, also called the regularisation term. Here, λ is a regularisation parameter and a value of less than 1 is recommended in [46]. The generator is now using random noise z and statistical information s to generate a representation that can be further described by substituting (9) described for our case in (11):

$$\min_G \max_D V_I(D, G) = \mathbb{E}[\log(I_n)] + \mathbb{E}[\log(1 - D(G(z, s)))] - \lambda I(s; G(z, s)), \tag{12}$$

which is basically subtraction of the mutual information (regularisation) term from the calculated loss in the GAN to improve the overall process of RL. A detailed description of the network implementing the InfoGAN is described in Table 4. The network uses the one-hot-code based latent code for disentangling the information of 20 classes of fruit and vegetables. Representation of each class is generated by a single hot-code to distinguish among the classes and random noise sampled by a normal distribution is conditioned based on this hot-code.

More recently, style based latent space disentanglement is reported in [57,71]. Compared to the previous efforts for improving discriminator networks [72–75], this work emphasises improvements in the overall generator architecture. The latent vectors considered as input are first disentangled with a non-linear dense network and added as a learned affine transformation in the input to control Adaptive Instance Normalisation (AdaIN). Our approach is largely based on a similar concept to enhance the dataset for ANN based fruit and vegetables classification. We have considered the texture details of images as the semantic code for further disentanglement of the latent as space discussed below.

Table 2 – Network description of 16-layer convolutional VAE.

Encoder			Decoder		
Layers	Input	Output		Input	Output
1 Conv2D	(3, 160, 160)	(16, 160, 160)	Linear	(1, 2 048)	(1, 2 048)
2 B.Norm2D	(16, 160, 160)	(16, 160, 160)	B.Norm	(1, 2 048)	(1, 2 048)
3 ReLU	(16, 160, 160)	(16, 160, 160)	ReLU	(1, 2 048)	(1, 2 048)
4 Conv2D	(16, 160, 160)	(32, 80, 80)	Linear	(1, 2 048)	(1, 25 600)
5 B.Norm2D	(32, 80, 80)	(32, 80, 80)	B.Norm	(1, 25 600)	(1, 25 600)
6 ReLU	(32, 80, 80)	(32, 80, 80)	ReLU	(1, 25 600)	(1, 25 600)
7 Conv2D	(32, 80, 80)	(64, 80, 80)	T.Conv2D	(1, 25 600)	(64, 80, 80)
8 B.Norm2D	(64, 80, 80)	(64, 80, 80)	B.Norm2D	(64, 80, 80)	(64, 80, 80)
9 ReLU	(64, 80, 80)	(64, 80, 80)	ReLU	(64, 80, 80)	(64, 80, 80)
10 Conv2D	(64, 80, 80)	(16, 40, 40)	T.Conv2D	(64, 80, 80)	(32, 80, 80)
11 B.Norm2D	(16, 40, 40)	(16, 40, 40)	B.Norm2D	(32, 80, 80)	(32, 80, 80)
12 ReLU	(16, 40, 40)	(16, 40, 40)	ReLU	(32, 80, 80)	(32, 80, 80)
13 Linear	(16, 40, 40)	(1, 2 048)	T.Conv2D	(32, 80, 80)	(16, 160, 160)
14 B.Norm1D	(1, 2 048)	(1, 2 048)	B.Norm2D	(16, 160, 160)	(16, 160, 160)
15 ReLU	(1, 2 048)	(1, 2 048)	ReLU	(16, 160, 160)	(16, 160, 160)
16 Linear	(1, 2 048)	(1, 2 048)	T.Conv2D	(16, 160, 160)	(3, 160, 160)

Conv2D: 2D Convolution, B.Norm2D: 2D Batch Normalisation, B.Norm1D: 1D Batch Normalisation

3.4. *Texture-based disentanglement*

Linear subspaces in a latent space provide significant control for RL in GANs. However, the probability of sampled latent code should correspond to the probability density of the actual dataset [57]. The same condition has been considered while generating the semantic code for IM in [46]. In a real world scenario, the datasets are not so disentangled and estimation of probability density for such sampling is a non-trivial task. Considering the new architecture of the generator described in [57,71], this constraint can be easily overlooked when, the latent space is unwrapped to linearise subsequent factors. A similar approach has been used in our case where the intermediate latent space $\mathscr{W}$ is similarly disentangled and combined with the texture information extracted from a subset of the training dataset. The images are processed with an eight layer dense ANN to estimate an initial semantic latent code for the 20 classes. The extracted texture details are compressed to a latent code with a maximum dimension matching the number of classes in the training dataset. This intermediate latent space is then used to learn the affine transforms for layer-by-layer style based representation learning. A description of the intermediate latent space generation process is illustrated in Fig. 6, where a single style

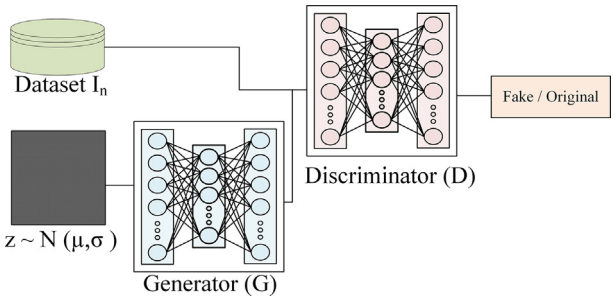


Fig. 5 – Basic architecture of a Generative Adversarial Network (GAN).

Table 3 – Network description of implemented Wasserstein GAN.

Layer	Generator	Discriminator
1	$z = \mathcal{N}(\mu,\sigma)\mu = 0, \sigma = 1$	I_n = (3, 160, 160)
2	Dense(100, 128)	Dense(784, 512)
3	LReLU(0.2)	LReLU(0.2)
4	Dense(128, 256)	Dense(512, 256)
5	B.Norm1D(256)	LReLU(0.2)
6	LReLU(0.2)	Linear(256, 1)
7	Dense(256, 512)	
8	B.Norm1D(512)	
9	LReLU(0.2)	
10	Dense(512, 1 024)	
11	B.Norm1D(1 024)	
12	LReLU(0.2)	
13	Dense(1 024, 784)	
14	Tanh()	

LReLU: Leaky ReLU, B.Norm1D: 1D Batch Normalisation

Table 4 – Network description of implemented InfoGAN.

Layer	Generator	Discriminator
1	$z = \mathcal{N}(\mu, \sigma)\mu = 0, \sigma = 1$	I_n = (3, 160, 160)
2	B.Norm2D(128, eps = $1e^{-5}$)	Conv2D(3, 3)
3	Upsample(s = 2.0, mode = nearest)	LReLU(0.2)
4	Conv2D(3, 3)	D.out2D(0.25)
5	B.Norm2D(128, eps = 0.8)	Conv2D(3, 3)
6	LReLU(0.2)	LReLU(0.2)
7	Upsample(s = 2.0, mode = nearest)	D.out2D(0.25)
8	Conv2D(3, 3)	B.Norm2D(32, eps = 0.8)
9	B.Norm2D(64, eps = 0.8)	Conv2D(3, 3)
10	LReLU(0.2)	LReLU(0.2)
11	Conv2D(3, 3)	D.out2D(0.25)
12	Tanh()	B.Norm2D(64, eps = 0.8)
13	LReLU: Leaky ReLU	Conv2D(3, 3)
14	B.Norm2D: 2D Batch Normalisation	LReLU(0.2)
15	D.out2D: 2D Dropout Layer	D.out2D(0.25)
16	Conv2D: 2D Convolution	B.Norm2D(128, eps = 0.8)

*The momentum for all 2D Batch Normalisation operations is 0.1.

block has been considered to describe the input as a learned affine transform and broadcast noise based on [71]. Each style block architecture is based on the latest state-of-the-art style based architecture described in [71]. The AdaIN operation [76] is broken into the constituent parts and demodulation instead of instance normalisation is used. Significant considerations related to better convergence and faster training have been made in our approach. The activation function is applied after addition of bias and broadcast noise in the style block. Previously LeakyReLU has been used in the original design, which has significant limitations when used for complex representation learning problems. Exponential Linear Unit (ELU) has provided a combination of the pros of both

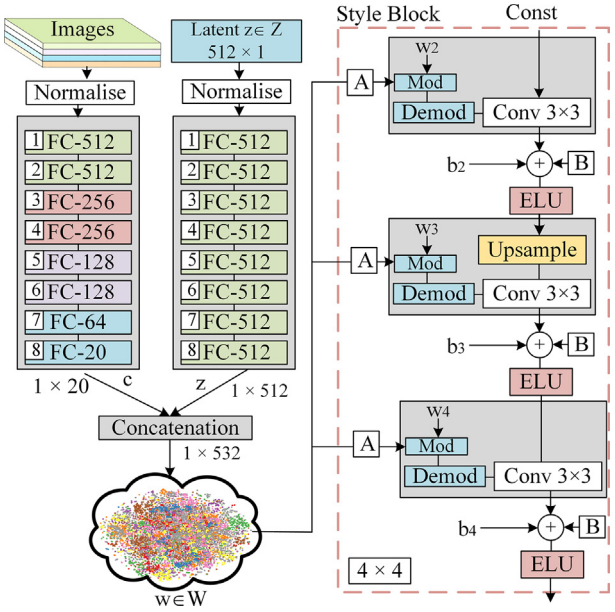


Fig. 6 – Building a texture-based intermediate latent space for a style based generative adversarial network,where)Aand)B represent the learned affine transform and broadcast noise, respectively.

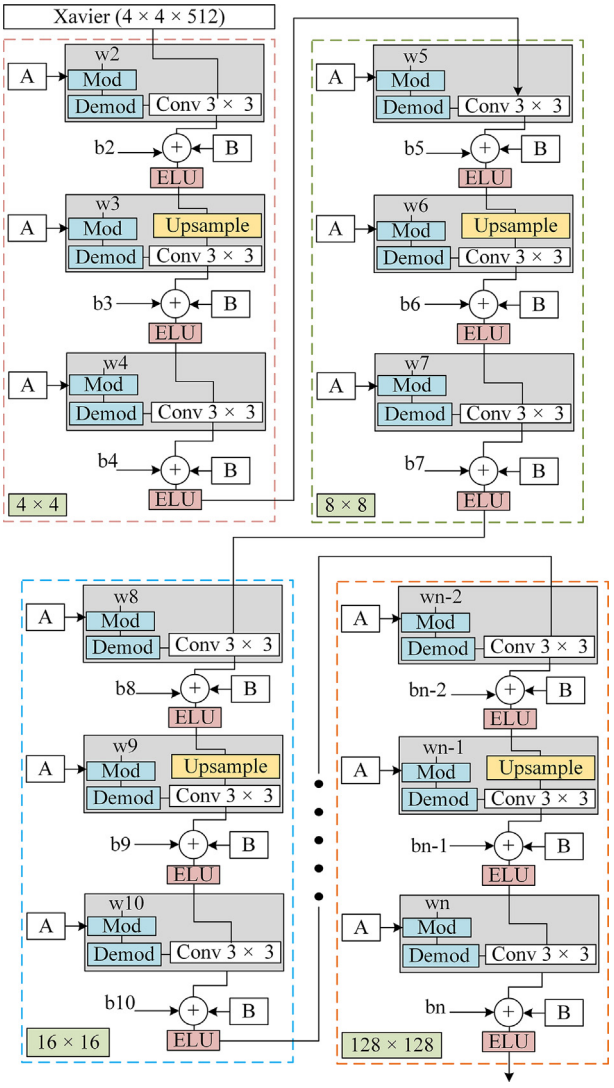


Fig. 7 – The network architecture for style transfer used in the proposed approach based on [55,71], where)Aand)B represent the learned affine transform and broadcast noise, respectively.

ReLU and LeakyReLU in [77]. The ELU improves the overall process of learning by producing the negative values and shifting the mean activation to zero, resulting in faster training. The complete network architecture of our approach has been presented in Fig. 7. The image synthesis process progressively increases in resolution (2^2–128^2) based on [55]. The process is initially started by considering a constant input of resolution 2^2. A normalised initialisation process [78] is used for this constant input and for the initial weights used in the overall training process, which reduces the difference of weight gradient and variance of network layers, hence providing a faster convergence rate.

We have further combined the concept of Information Maximisation (IM) with the texture-based latent space disentanglement. The shared high level information between data elements can be learned by enhancing the mutual information between them, where a significant challenge in this regard is the reduction of low level noise. The initial latent code estimated by passing images through the mapping network reduces low level noise while preserving the abstract texture details. The IM technique used in our approach improves the control over the latent code in the image synthesis. The initial latent code is improved based on the mutual information maximisation principle defined in (11). The implementation of the IM principle in our described approach is based on [46], where the auxiliary distribution for lower bound estimation is obtained by adding a dense layer in the discriminator based on the continuous nature of the semantic latent code and the auxiliary distribution is treated as a factorised Gaussian for estimation of mutual IM. A complete description of the implemented generator and discriminator architecture is described in Tables 5 and Table 6 for the mapping and synthesis networks, respectively.

The image dataset has been divided for training and semantic latent code estimation. A disjoint set of 640 images (20% of dataset) per class have been used for semantic latent estimation, where 2 560 (80% of dataset) images per class have been used for training the texture-based style representation learning.

The implemented ANN is considered a form of Helmholtz machine, while the algorithm is defined in [79] as phases of synthesis and information estimation for the training process. A similar rule has been used for training the defined approach. The synthesis phase draws the image samples for estimation of auxiliary distribution and the information estimation phase updates the auxiliary distribution for better representation. The assumption behind this implementation is that combining the pros from the techniques presented in [46,55,57,71] will result in better RL outcomes incorporating the stochastic nature of real-life images.

4. Results

The results of the above implemented techniques for our dataset of supermarket fruit and vegetables have been illustrated in the form of images, Frechet Inception Distance (FID) and classification accuracy between groundtruth and synthesised images as a statistical metric. The FID and classification accuracy are considered as statistical metrics to estimate the effectiveness of the proposed technique. These statistical metrics are considered based on the state-of- the-art techniques used for synthesised image analysis. More details on both techniques used can be found in Sections 4.2 and 4.3.

4.1. *Synthesised images*

A single layer dense VAE is implemented for 256, 512 and 1 024 nodes to analyse the capability of RL for a higher number of nodes in a single layer. An assumption of single layer ANN convergence to any arbitrary function is considered for implementation of the dense VAE. The network described in (2) is trained on 51 200 random images of 20 classes with a batch size of 32 for 64 epochs, while 12 800 images have been considered as a random disjoint set to test the representation effectiveness of the network. The adaptive learning rate of individual parameters [80] has been used for optimisation of the network, where the loss has been estimated using the Mean Squared Error (MSE). A set of example groundtruth and generated images is shown in Fig. 8. The spatial organisation of colour and edges can be extracted with a convolutional operation. A convolutional VAE network is described in Table 2 to consider the texture details for the RL process. The network is trained using our dataset to learn a representation for fruit and vegetables. The adaptive learning rate of individual parameters is optimised to improve the effectiveness of RL. The results are presented in Fig. 9 for groundtruth images and generated images. More detailed colour and less blurred texture information can be observed in the images generated with convolutional VAE. Even deeper networks could be used

Table 5 – Description of synthesis network for texture-based intermediate latent synthesis using groundtruth images and 512-dimensional noise.

	Mapping (Texture)	Mapping (Noise)
1	Images (3, 128, 128)	$z = \mathcal{N}(\mu, \sigma)\mu = 0, \sigma = 1$
2	Dense (16 384, 512)	Dense (1, 512)
3	Dense(512, 512)	Dense (512, 512)
4	Dense(512, 256)	Dense (512, 512)
5	Dense(256, 256)	Dense (512, 512)
6	Dense(256, 128)	Dense (512, 512)
7	Dense(128, 128)	Dense (512, 512)
8	Dense(128, 64)	Dense (512, 512)
9	Dense(64, 20)	Dense (512, 512)

Table 6 – Detailed architectures of generator and discriminator for synthesis neural networks.

Generator architecture Output size Weight matrix				Discriminator architecture Output size Weight matrix		
Layesrs		Output size	Weight matrix	Layers	Output size	Weight matrix
1	$Input_{latent}$	(1, 532)	–	$Image_{in}$	(3, 128, 128)	–
2	$Const_{4\times4}$	(512, 4, 4)	(1, 512, 4, 4)	$FromRGB_{128\times128}$	(256 , 128, 128)	(1, 1, 3, 256)
3	$Conv_{4\times4}$	(512, 4, 4)	(3, 3, 512, 512)	$Conv_{128\times128}$	(256, 128, 128)	(3, 3, 256, 256)
4	$ToRGB_{4\times4}$	(3, 4, 4)	(1, 1, 512, 3)	$ConvDown_{128\times128}$	(512, 64, 64)	(3, 3, 256, 512)
5	$ConvUp_{8\times8}$	(512, 8, 8)	(3, 3, 512, 512)	$SkipConn_{128\times128}$	(512, 64, 64)	(1, 1, 256, 512)
6	$Conv_{8\times8}$	(512, 8, 8)	(3, 3, 512, 512)	$Conv_{64\times64}$	(512, 64, 64)	(3, 3, 512, 512)
7	$Upsample_{8\times8}$	(3, 8, 8)	–	$ConvDown_{64\times64}$	(512, 32, 32)	(3, 3, 512, 512)
8	$ToRGB_{8\times8}$	(3, 8, 8)	(1, 1, 512, 3)	$SkipConn_{64\times64}$	(512, 32, 32)	(3, 3, 512, 512)
9	$ConvUp_{16\times16}$	(512, 16, 16)	(3, 3, 512, 512)	$Conv_{32\times32}$	(512, 32, 32)	(3, 3, 512, 512)
10	Conv16 × 16	(512, 16, 16)	(3, 3, 512, 512)	$ConvDown_{32\times32}$	(512, 16, 16)	(3, 3, 512, 512)
11	$Upsample_{16\times16}$	(3, 16, 16)	–	$SkipConn_{32\times32}$	(512, 16, 16)	(1, 1, 512, 512)
12	$ToRGB_{16\times16}$	(3, 16, 16)	(1, 1, 512, 3)	$Conv_{16\times16}$	(512, 16, 16)	(3, 3, 512, 512)
13	$ConvUp_{32\times32}$	(512, 32, 32)	(3, 3, 512, 512)	$ConvDown_{16\times16}$	(512, 8, 8)	(3, 3, 512, 512)
14	$Conv_{32\times32}$	(512, 32, 32)	(3, 3, 512, 512)	$SkipConn_{16\times16}$	(512, 8, 8)	(1, 1, 512, 512)
15	$Upsample_{32\times32}$	(3, 32, 32)	–	$Conv_{8\times8}$	(512, 8, 8)	(3, 3, 512, 512)
16	$ToRGB_{32\times32}$	(3, 32, 32)	(1, 1, 512, 3)	$ConvDown_{8\times8}$	(512, 4, 4)	(3, 3, 512, 512)
17	$ConvUp_{64\times64}$	(512, 64, 64)	(3, 3, 512, 512)	$SkipConn_{8\times8}$	(512, 4, 4)	(1, 1, 512, 512)
18	$Conv_{64\times64}$	(512, 64, 64)	(3, 3, 512, 512)	$Mini\text{-}Stddev_{4\times4}$	(513, 4, 4)	–
19	$Upsample_{64\times64}$	(3, 64, 64)	–	$Conv_{4\times4}$	(512, 4, 4)	(3, 3, 513, 512)
20	$ToRGB_{64\times64}$	(3, 64, 64)	(1, 1, 512, 3)	$Dense_{4\times4}$	(512)	(8 192, 512)
21	$ConvUp_{128\times128}$	(256, 128, 128)	(3, 3, 512, 256)	Output	(1)	(512, 1)
22	$Conv_{128\times128}$	(256, 128, 128)	(3, 3, 256, 256)	$Score_{Out}$	(1)	–
23	$Upsample_{128\times128}$	(3, 128, 128)	–			
24	$ToRGB_{128\times128}$	(3, 128, 128)	(1, 1, 256, 3)			
25	$Images_{out}$	(3, 128, 128)	–			

ToRGB: RGB Conversion, ConvUp: Up Convolution, Upsample: Up Sampling Layer.

to extract more abstract information of texture and colour. However, deeper convolutional networks also have limitations w.r.t. model size, training and synthesis time.

GANs are the most prominent network in the state-of-the-art techniques for representation learning for data enhancement, data augmentation and representation based classifi-

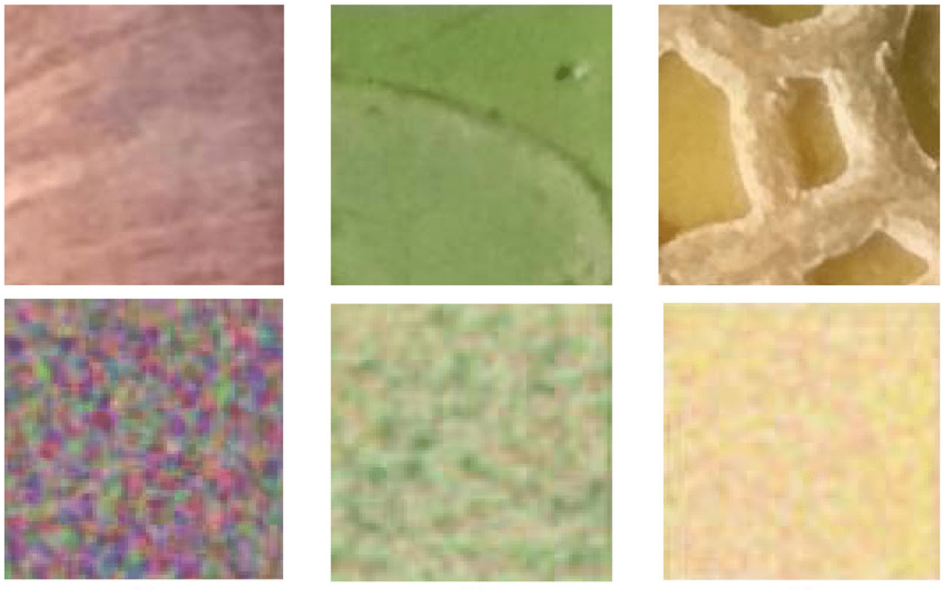


Fig. 8 – Comparison of results for Dense VAE: (a) image reconstruction of Brown Onion with 256 nodes, (b) image reconstruction of Iceberg Lettuce with 512 nodes, and (c) image reconstruction of Rockmelon with 1 024 nodes.

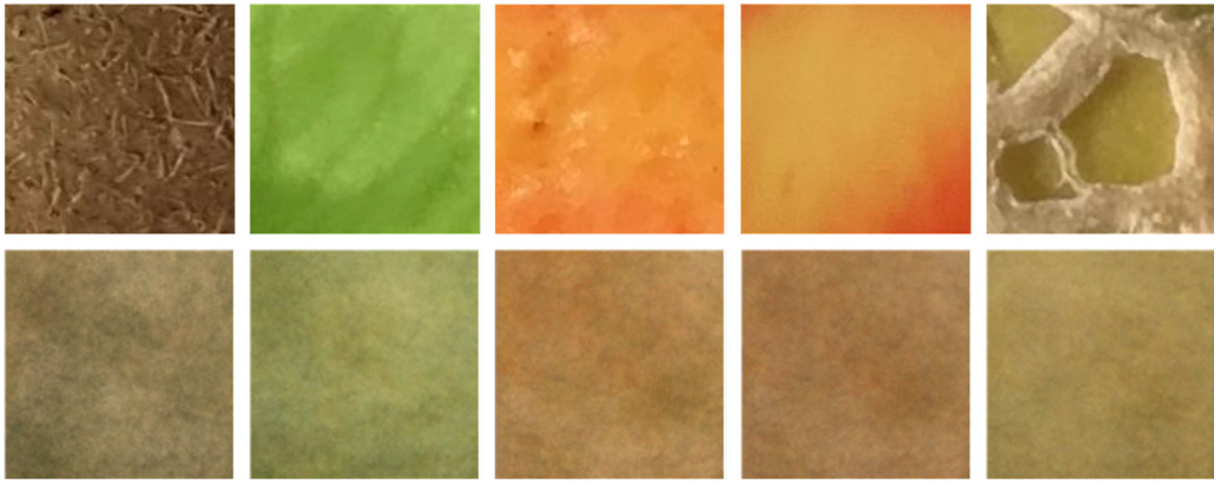


Fig. 9 – Comparison of example groundtruth images (upper row) and synthesised images (lower row) of Kiwifruit, Iceberg Lettuce, Orange, Red Delicious Apple and Rockmelon, respectively for Convolutional VAE after 3 200 iterations (left to right).

cation problems. Two kinds of GAN variants have been considered for this implementation based on the recent significant methodology improvements in GAN design. The vertical difference based cost has been evaluated for the VAEs and the initial GAN designs using KL and JS based distribution divergence. The KL and JS divergence have an inherent limitation of having constant cost for disjoint distributions, however, many real-life complex applications of GANs can have disjoint distributions. This issue has been resolved in Wasserstein GANs by using horizontal distance as a distance metric. The Wasserstein GANs can model the disjoint distribution divergence, which can be used for better parameter updating for generator and discriminator. The Wasserstein GANs based RL for fruit and vegetables is presented in Fig. 10, Fig. 11 and Fig. 12 sampled at 1 600, 3 200 and 6 400 iterations, respectively. The mutual information maximisation based InfoGAN has been implemented to restrict the random distribution conditioned on statistical semantic information for RL.

The information maximisation is added as a regularisation term to improve the GAN objective function which can result in faster and more effective RL for complex problems. Both networks have been trained using the fruit and vegetables dataset with a batch size and epoch size of 64, where Adam optimisation [80] is used for parameter optimisation. A comparison of the groundtruth and InfoGAN based generated images is presented in Fig. 13. Unwrapping of latent space to linearise subsequent components has been reported as a significant effort to control the stochastic nature of the synthesis process in GANs [46,57,71]. A texture-based disentanglement has been implemented with a presumption of synthesising more texture details in images to enhance the dataset for ANN based classification of fruit and vegetables as an application. Example images have been presented in Fig. 14.

4.2. *Statistical metrics*

The mean and co-variance difference of high dimensional image data distributions can be estimated by the Freet Inception Distance (FID) [81]. This distance can be used as a statis-

Fig. 10 – Comparison of results for Wasserstein GAN example groundtruth images (upper row) and synthesised images (lower row) after 1 600 iterations for Honeydew Melon, Kiwifruit, Packham Pear, Topless Pineapple and Red Delicious Apple (left to right).

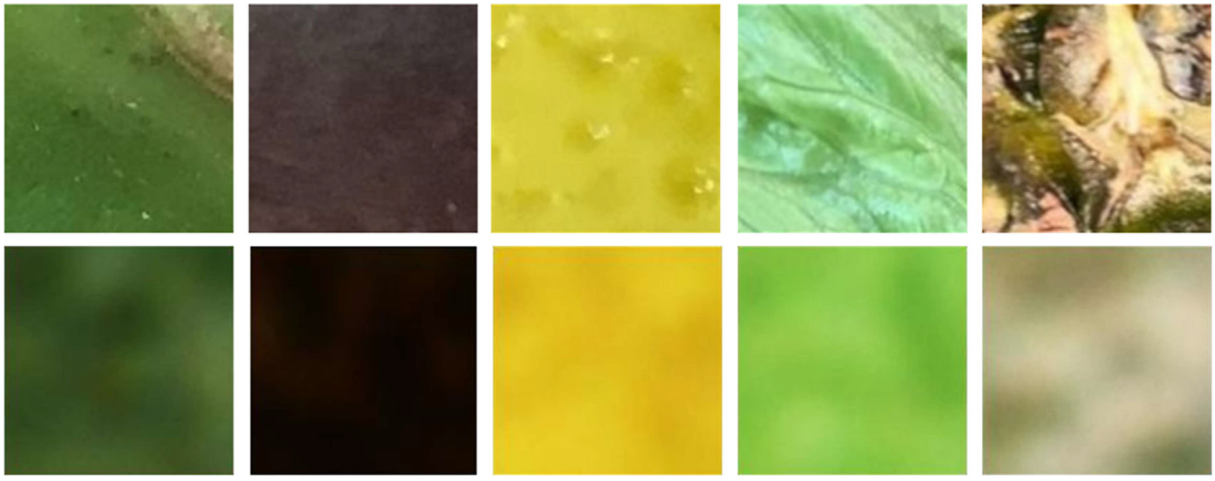


Fig. 11 – Comparison of results for Wasserstein GAN example groundtruth images (upper row) and synthesised images (lower row) after 3 200 iterations for Capsicum, Eggplant, Lemon, Iceberg Lettuce, Topless Pineapple (left to right).

Fig. 12 – Comparison of results for Wasserstein GAN example groundtruth images (upper row) and synthesised images (lower row) after 6 400 iterations for Eggplant, Kiwifruit, Orange, Topless Pineapple, Red Delicious Apple (left to right).

tical measure for synthesised image quality estimation [82]. To estimate this distance, a high dimensional distribution of 2 048 feature activations learned by the penultimate layer of Inception V3 [83] for both the synthesised and groundtruth images are obtained. These activation feature vectors are then used to estimate the following FID:

$$d^2\{(\mu, C_{ov}), (\mu_s, C_{ovs})\} = \|\mu - \mu_s\|_2^2 + T_r\left(C_{vs} + C_{ovs} - 2(C_{vs}C_{ovs})^{\frac{1}{2}}\right), \quad (13)$$

where μ, μ_s and C_{vs}, C_{ovs} represent the feature based mean and co-variance of the feature activation vectors of groundtruth and synthesised images, respectively, and the trace matrix is represented by T_r. The FID estimation of synthesised images w.r.t. groundtruth images is described in Table 7. The quantitative comparison between the synthesised images and the training dataset illustrates significant overall image quality enhancement due to the improvements resulting from better distribution divergence and the latent space disentanglement techniques. As FID is based on Inception V3 [83] which has recently been proposed for better texture-based image classification, a lower FID indicates better image similarity and with significant texture details. The synthesised images with lower FID can be considered as novel samples, and hence can be used for training a CNN for texture and colour based classification of fruit and vegetables. The evaluation and comparison of images generated with the help of VAEs and GANs has significant implications for future research directions. As VAEs and GANs are purely unsupervised techniques, better representations can be learned with the concept of IM and enhanced disentanglement as a semi-supervised approach. The concept of a better distance metric is also useful for disjoint distribution divergence estimation. More sophisticated statistical semantics information

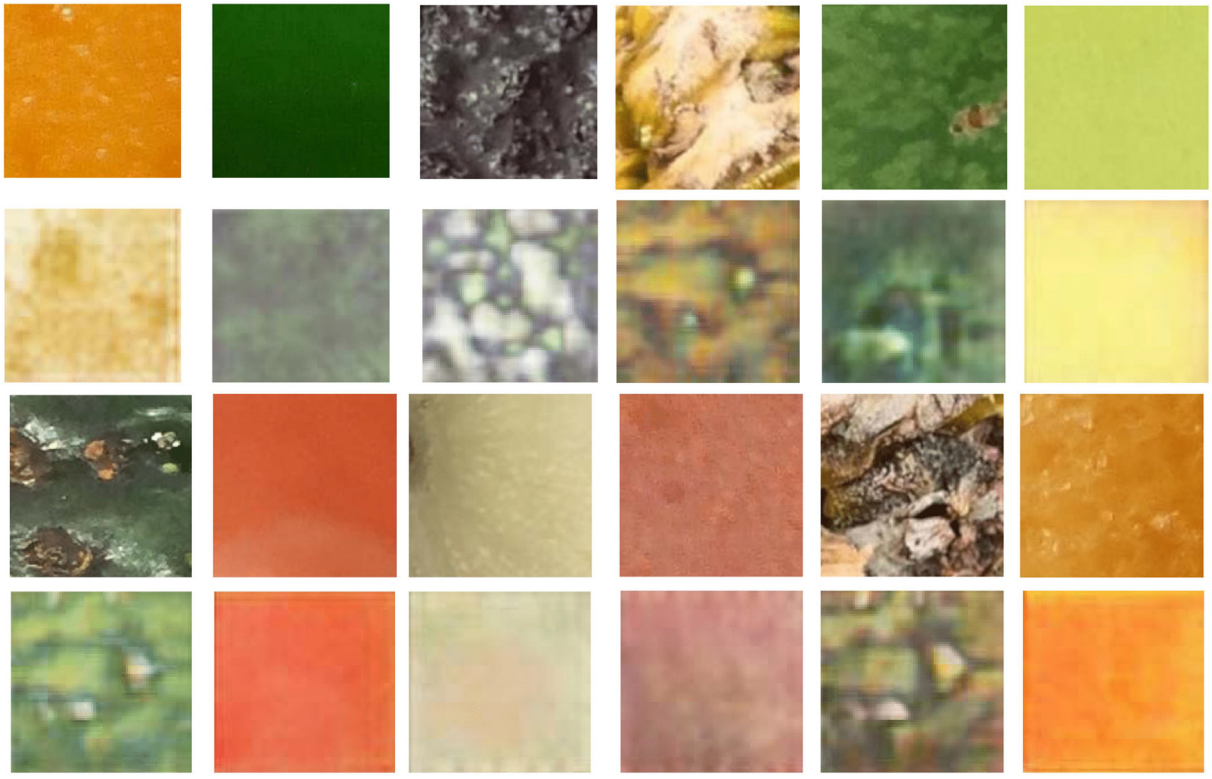


Fig. 13 – Comparison of results for InfoGAN: (a) samples after 16 000 iterations for Orange, Capsicum and Hass Avocado, (b) samples after 32 000 iterations for Topless Pineapple, Watermelon and Yellow Banana, (c) samples after 48 000 iterations for Hass Avocado, Tomato and Honeydew Melon, and (d) samples after 64 000 iterations for Potato, Topless Pineapple and Orange (left to right).

can be helpful to improve the overall process of RL for fruit and vegetables dataset enhancement and classification.

4.3. *Classification accuracy*

A DCNN based image classification is performed to estimate the image data enhancement. A pre-trained ResNet50 [84] on ImageNet [85] is transfer learned with the groundtruth and synthesised images for classification. The initial 45-layers are considered as the bottleneck feature extractor where the last five layers are adapted considering the hypothesis of significant disparity between ImageNet and our classification task. This technique can also achieve significant auxiliary goals of computational cost reduction and overfitting. The extracted features by bottleneck and adapted layers are then used to train a softmax classifier with 20 classes where a categorical cross entropy and Adam Optimiser are employed as loss function and optimiser, respectively. The weights for trainable layers are initialised using Xavier uniform initialisation. A 10-fold cross validation with random sample shuffling using an approximate proportion with class labels is used for training the ResNet.

Training accuracy and loss for groundtruth and synthesised images is illustrated in Fig. 15(a). The ResNet is transfer learned for the groundtruth dataset and synthesised images separately on 51 200 and 3 200 images, respectively. The testing has been performed on a disjoint image dataset. A comparison of average test accuracy and error for both groundtruth and synthesised images is depicted in Fig. 15(b) and Fig. 15(c). The test set for both datasets is portioned in three disjoint sets of 10 images per class where accuracy is estimated for each class with transferred learned ResNet(s). The per class accuracy obtained by disjoint test sets is averaged to find the mean accuracy and error range for both the groundtruth and synthesised images. A significant conjunction between the range of the estimated accuracy represented by the error bar of multiple classes implicates the dataset similarities and, hence, dataset enhancement. Smaller difference between the mean accuracy of the groundtruth and synthesised images can also be used to estimate the similarity of the two training datasets used. A much larger dataset can be built and used for training complex and deeper DCNNs. Significant similarity as a smaller mean accuracy difference can be observed for more texture rich fruit and vegetable images.

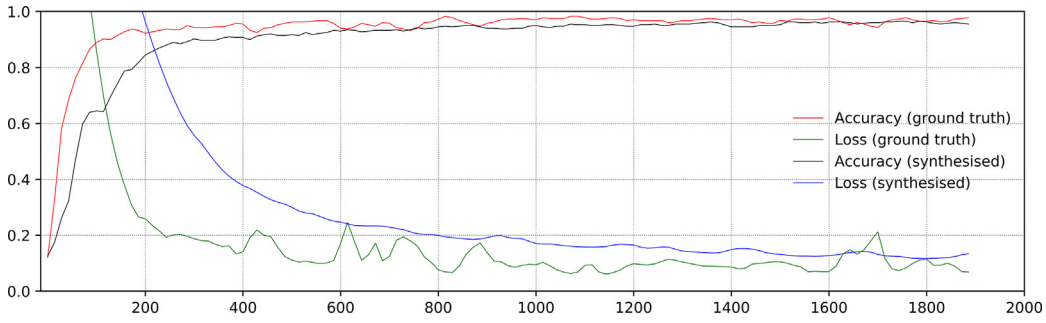
5. Conclusion

This paper has evaluated different Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs) and texture-based latent space disentanglement based GAN for the state-of-the-art application of Representation Learning (RL) of a fruit and vegetables dataset. The scarcity of large labelled datasets of fruit and vegetables is the major motiva-

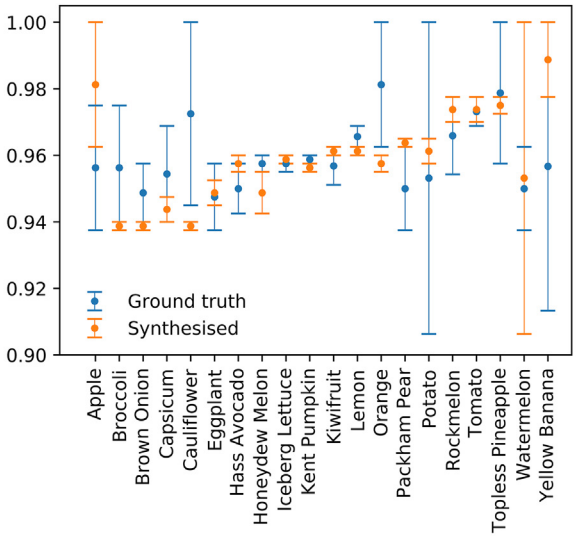
Fig. 14 – Example synthesised images based on the texture-based latent space disentanglement for dataset enhancement.

Table 7 – Frechet Inception Distance (FID) estimation of synthesised and groundtruth images.

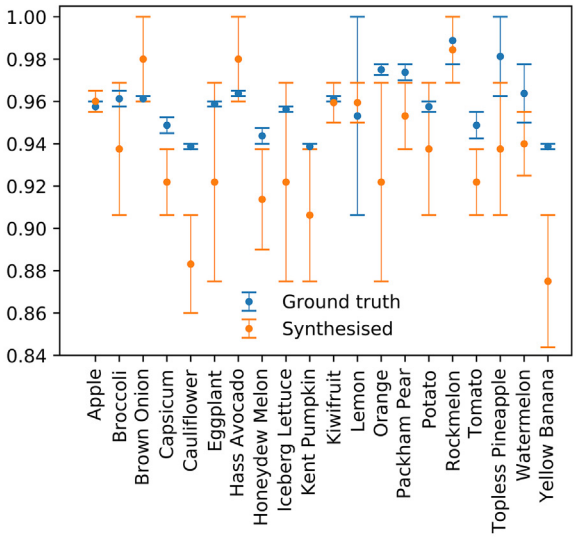
	Method	FID Estimation
1	Dense VAE (256)	46.23
2	Dense VAE(512)	48.56
3	Dense VAE(1 024)	44.15
4	Convolutional VAE	34.56
5	Wasserstein GAN(1 600)	34.78
6	Wasserstein GAN(3 200)	32.67
7	Wasserstein GAN (6 400)	31.33
8	InfoGAN (16 000)	22.45
9	InfoGAN(32 000)	21.78
10	InfoGAN(48 000)	19.99
11	InfoGAN(64 000)	16.74
12	Our Proposed Approach	5.18



(a)



(b)



(c)

Fig. 15 – The ResNet based classification accuracy estimation for groundtruth and synthesised images: (a) Transfer learning accuracy and loss, (b) A comparison of mean classification accuracy and error (transfer learned on groundtruth), and (c) A comparison of mean classification accuracy and error (transfer learned on synthesised images).

tion to perform RL for data enhancement. The classification of fruit and vegetables is a multi-class classification problem with significant inherent limitations. Convolutional Neural Networks (CNNs) are a more suitable technique for such multi class classification tasks, however the effective training of CNNs requires a significantly large labelled dataset. The VAEs and GANs have been evaluated to use the learned representation for novel sample generation. This representation can also be extended for feature extraction for classification of fruit and vegetables, which is the main purpose of this study.

Comparison of the implemented VAEs and GANs shows that using the Wasserstein distance as a distribution divergence metric improve the results significantly as compared to dense VAE and convolutional VAE. Colour and texture details are more evident in the test images generated with the help of a Wasserstein GAN whereas the images generated with dense VAE are similar to colour noise. The images generated with convolution VAE have no discrete texture and colour information that can be used for classification or training of a supervised CNN. The mutual information maximisation used in the Information Maximisation Generative Adversarial Network (InfoGAN) further improves representation and distinct colour and texture information is evident in the generated images. As evident from the results a combination of mutual information maximisation and use of the Wasserstein distance metric can improve the performance significantly. The transformation of a high dimensional latent space to linear sub spaces can provide significant control over the representation learning and synthesis process. The disentanglement obtained by the linear sub space transformation has been explored to achieve more effective results with texture details. We have used a combination of disentanglement techniques to achieve our goal of dataset enhancement for Artificial Neural Network (ANN) based fruit and vegetables classification. The texture details of a subsequent random disjoint subset of training data has been used as a semantic latent code to control the use of disentangled latent space for image synthesis in the generator network. The image size has been limited to 128×128 resolution to reduce the time and computational complexity, however the results illustrate that more texture rich high resolution images can also be synthesised using the same approach.

As part of our future work, we will investigate sophisticated statistical semantic information for better disentanglement and use with semi supervised GANs [86]. There is a potential of improvement using semi-supervised GANs with small labelled datasets of fruit and vegetables. We will also investigate more sophisticated distance metrics and a combination of different distance metrics with sophisticated semantic information to improve RL for fruit and vegetables. The learned representation can be used for generation of novel samples and feature extraction for classification as our final goal.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported by Edith Cowan University (ECU), Australia and Higher Education Commission (HEC) Pakistan, The Islamia University of Bahawalpur (IUB) Pakistan (5-1/HRD/UE STPI(Batch-V)/1182/2017/HEC). The authors would like to thank ECU Australia, HEC and IUB Pakistan for the PhD scholarship granted to the first author of this paper.

REFERENCES

[1] Zhang Y, Wang S, Ji G, Phillips P. Fruit classification using computer vision and feedforward neural network. J Food Eng 2014;143:167–77.
[2] Zhang Y-D, Dong Z, Chen X, Jia W, Du S, Muhammad K, et al. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. Multimedia Tools Appl 2019;78(3):3613–32.
[3] Nasirahmadi A, Miraei Ashtiani SH. Bag-of-Feature model for sweet and bitter almond classification. Biosyst Eng 2017;156 (4):51–60.
[4] Behera SK, Sangita S, Rath AKSP. Automatic classification of mango using statistical feature and SVM. Adv Comput Commun Control Lecture Notes Netw Syst 2019;41:469–75.
[5] Tripathi MK, Maktedar DD. A role of computer vision in fruits and vegetables among various horticulture products of agriculture fields: a survey. Inform Process Agric 2019;7 (2):183–203.
[6] Rehman TU, Mahmud MS, Chang YK, Jin J, Shin J. Current and future applications of statistical machine learning algorithms for agricultural machine vision systems. Comput Electron Agric 2019;156:585–605.
[7] Bhargava A, Bansal A. Fruits and vegetables quality evaluation using computer vision: a review. J King Saud Univ – Comput Inform Sci 2021;33(3):243–57.
[8] Moallem P, Serajoddin A, Pourghassem H. Computer vision-based apple grading for golden delicious apples based on surface features. Inform Process Agric 2017;4(1):33–40.
[9] Nouri-Ahmadabadi H, Omid M, Mohtasebi SS, Soltani Firouz M. Design, development and evaluation of an online grading system for peeled pistachios equipped with machine vision technology and support vector machine. Inform Process Agric 2017;4(4):333–41.
[10] Arakeri MP, Lakshmana. Computer vision based fruit grading system for quality evaluation of tomato in agriculture industry. Proc Comput Sci 2016;79:426–33.
[11] Jhawar J. Orange sorting by applying pattern recognition on colour image. Procedia Comput Sci 2016;78:691–7.
[12] Sofu MM, Er O, Kayacan MC, Ceti3li B. Design of an automatic apple sorting system using machine vision. Comput Electron Agric 2016;127:395–405.
[13] Mahendran R, Gc J, Alagusundaram K. Application of computer vision technique on sorting and grading of fruits and vegetables. J Food Process Technol 2012;10(1):2157–65.
[14] Al Ohali Y. Computer vision based date fruit grading system: design and implementation. J King Saud Univ – Comput Inform Sci 2011;23(1):29–36.
[15] Liming X, Yanchao Z. Automated strawberry grading system based on image processing. Comput Electron Agric 2010;71 (1):32–9.
[16] Hameed K, Chai D, Rassau A. A comprehensive review of fruit and vegetable classification techniques. Image Vis Comput 2018;80:24–44.

[17] Hameed K, Chai D, Rassau A. A progressive weighted average weight optimisation ensemble technique for fruit and vegetable classification. In: Proc of the 16th international conference on control, automation, robotics and vision, ICARCV. Shenzhen, China; 2020. p. 303–8.
[18] Hameed K, Chai D, Rassau A. A sample weight and adaboost cnn-based coarse to fine classification of fruit and vegetables at a supermarket self-checkout. Appl Sci 2020;10(23):8667.
[19] Hussain Hassan NM, Nashat AA. New effective techniques for automatic detection and classification of external olive fruits defects based on image processing techniques. Multidimension Syst Signal Process 2019;30(2):571–89.
[20] Habib MT, Majumder A, Jakaria AZM, Akter M, Uddin MS, Ahmed F. Machine vision based papaya disease recognition. J of King Saud Univ – Comput Inform Sci 2020;32(3):300–9.
[21] Braverman M. Polylogarithmic independence fools AC0 circuits. J Assoc Comput Mach 2008;57(5):10–25.
[22] Sun C, Shrivastava A, Singh S, Gupta A. Revisiting unreasonable effectiveness of data in deep learning era. In: Proc of the 15th IEEE international conference on computer vision, ICCV. Venice, Italy; 2017. p. 843–52.
[23] Bengio Y, Delalleau O, Roux NL. The curse of highly variable functions for local kernel machines. In: Proc of the 19th international conference on neural information processing systems, NIPS. Vancouver, Canada; 2005. p. 107–14.
[24] Bottou L. From machine learning to machine reasoning: an essay. J Mach Learn Res 2013;94(2):3207–60.
[25] Takahashi R, Matsubara T, Uehara K. Data augmentation using random image cropping and patching for deep CNNs. IEEE Trans Circ Syst Video Technol 2019;30(9):2917–31.
[26] Yang Q, Pan SJ. A survey on transfer learning. IEEE Trans Knowl Data Eng 2009;22(10):1345–59.
[27] Kingma DP, Welling M. Auto-encoding variational bayes. In: Proc of the 2nd international conference on learning representations, ICLR. Banff, AB, Canada; 2014. p. 214–28.
[28] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al.. Generative adversarial nets. In: Proc of the 35th neural information processing systems, NIPS. Montreal, Canada; 2014. p. 2672–80.
[29] Theis L, van den Oord A, Bethge M. A note on the evaluation of generative models. In: Proc of the 4th international conference on learning representations, ICLR. San Juan, Puerto Rico; 2016. p. 25–35.
[30] Jalwana MAAK, Akhtar N, Bennamoun M, Mian A. Attack to explain deep representation. In: Proc of the IEEE conference on computer vision and pattern recognition CVPR. Seattle, WA, USA; 2020. p. 9540–9.
[31] Shah RA, Qian Y, Kumar D, Ali M, Alvi MB. Network intrusion detection through discriminative feature selection by using sparse logistic regression. Future Internet 2017;9(4):81.
[32] Alzahrani AS, Shah RA, Qian Y, Ali M. A novel method for feature learning and network intrusion classification. Alexandria Eng J 2020;59(3):1159–69.
[33] Zardari ZA, Memon KA, Shah RA, Dehraj S, Ahmed I. A lightweight technique for detection and prevention of wormhole attack in manet. EAI Endorsed Trans Scalable Inform Syst 2020;8(29):1–6.
[34] Bousmalis K, Silberman N, Dohan D, Erhan D, Krishnan D. Unsupervised pixel-level domain adaptation with generative adversarial networks. In: Proc of the IEEE conference on computer vision and pattern recognition, CVPR. Honolulu, Hawaii, USA; 2017. p. 95–104.
[35] Le Roux N, Bengio Y. Representational power of restricted boltzmann machines and deep belief networks. Neural Comput 2008;20(6):1631–49.
[36] Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. Neural Comput 2006;18(7):1527–54.
[37] Rezende DJ, Mohamed S, Wierstra D. Stochastic backpropagation and approximate inference in deep generative models. In: Proc of the 31st international conference on machine learning, ICML. Beijing, China; 2014. p. 3057–70.
[38] Van Den Oord A, Kalchbrenner N, Kavukcuoglu K. Pixel recurrent neural networks. In: Proc of the 33rd international conference on machine learning, ICML. New York, USA; 2016. p. 2611–20.
[39] Bengio Y, Thibodeau-Laufer É, Alain G, Yosinski J. Deep generative stochastic networks trainable by backprop. In: Proc of the 31st international conference on machine learning, ICML. Beijing, China; 2014. p. 1470–85.
[40] Van Den Oord A, Kalchbrenner N, Vinyals O, Espeholt L, Graves A, Kavukcuoglu K. Conditional image generation with PixelCNN decoders. In: Proc of the 30th conference on neural information processing systems, NIPS. Barcelona, Spain; 2016. p. 4797–805.
[41] Arjovsky M, Bottou L. Towards principled methods for training generative adversarial networks. In: Proc. of the 5th international conference on learning representations, ICLR. Toulon, France; 2017. p. 1355–72.
[42] Che T, Li Y, Jacob AP, Bengio Y, Li W. Mode regularized generative adversarial networks. In: Proc of the 5th international conference on learning representations, ICLR. Toulon, France; 2017. p. 82–6.
[43] Ghosh A, Kulharia V, Namboodiri V, Torr PH, Dokania PK. Multi-agent diverse generative adversarial networks. In: Proc of the IEEE conference on computer vision and pattern recognition, CVPR. Utah, USA; 2018. p. 8513–21.
[44] Salimans Tim, Goodfellow Ian, Zaremba Wojciech, Cheung Vicki, Radford Alec, Chen X. Improved techniques for training GANs. In: Proc of the 30th conference on neural information processing systems, NIPS. Barcelona, Spain; 2016. p. 2234–42.
[45] Yan X, Yang J, Sohn K, Lee H. Attribute2image: Conditional image generation from visual attributes. In: Proc of the 14th European conference on computer vision ECCV. Amsterdam, Netherlands; 2016. p. 776–91.
[46] Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Proc of the 30th conference on neural information processing systems, NIPS. Barcelona, Spain; 2016. p.2172–80.
[47] Zhu JY, Krähenbühl P, Shechtman E, Efros AA. Generative visual manipulation on the natural image manifold. In: Proc of the 14th European conference on computer vision, ECCV. Amsterdam, Netherlands; 2016. p. 597–613.
[48] Isola P, Zhu JY, Zhou T, Efros AA. Image-to-image translation with conditional adversarial networks. In: Proc of the IEEE conference on computer vision and pattern recognition, CVPR. Honolulu, Hawaii, USA; 2017. p. 1125–34.
[49] Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A et al. Photo-realistic single image super-resolution using a generative adversarial network. In: Proc of the IEEE conference on computer vision and pattern recognition CVPR. Honolulu, Hawaii, USA; 2017. p. 105–14.
[50] Zhang H, Xu T, Li H, Zhang S, Wang X, Huang X, et al. StackGAN++: realistic image synthesis with stacked generative adversarial networks. IEEE Trans Pattern Anal Mach Intell 2019;41(8):1947–62.
[51] Mansimov E, Parisotto E, Ba LJ, Salakhutdinov R. Generating images from captions with attention. In: Proc of the 4th international conference on learning representations, ICLR. San Juan, Puerto Rico; 2016. p. 40–8.
[52] Nguyen A, Clune J, Bengio Y, Dosovitskiy A, Yosinski J. Plug & play generative networks: conditional iterative generation of images in latent space. In: Proc of the IEEE conference on

computer vision and pattern recognition, CVPR. Honolulu, Hawaii, USA; 2017. p. 4467–77.
[53] Wang X, Gupta A. Generative image modelling using style and structure adversarial networks. In: Proc of the 14th European conference on computer vision, ECCV. Amsterdam, Netherlands; 2016. p. 318–35.
[54] Denton EL, Chintala S, Fergus R, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In: Proc of the 29th international conference on neural information processing systems, NIPS. Montreal, Canada; 2015. p. 1486–94.
[55] T Karras, T Alia, SLiane J. Progressively growing of GANs for improved quality, stability and variation. In: Proc of the 6th international conference on learning representations, ICLR. Vancouver, Canada; 2018. p. 50–6.
[56] Desjardins G, Courville A, Bengio Y. Disentangling factors of variation via generative entangling. In: Proc of the 26th neural information processing systems, NIPS. Lake Tahoe, USA; 2012. p. 5040–8.
[57] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. In: Proc of the IEEE conference on computer vision and pattern recognition, CVPR. Long Beach, CA, USA; 2019. p. 4401–10.
[58] Hénaff OJ, Razavi A, Doersch C, Eslami S, Oord Avd. Data-efficient image recognition with contrastive predictive coding. In: Proc of the 8th international conference on learning representations, ICLR. Addis Ababa, Ethiopia; 2020. p. 1–10.
[59] Van den Oord A, Li Y, Vinyals O. Representation learning with contrastive predictive coding. Comput Res Repos 2018;1807.03748(1):1–13.
[60] Zhang R, Isola P, Efros AA. Colorful image colorization. In: Proc of the 14th European conference on computer vision, ECCV. Amsterdam, Netherlands; 2016. p. 649–66.
[61] Pu Y, Gan Z, Henao R, Yuan X, Li C, Stevens A, Carin L. Variational autoencoder for deep learning of images, labels and captions. In: Proc of the 30th conference on neural information processing systems, NIPS. Barcelona, Spain; 2016. p. 2360–68.
[62] Zeiler MD, Taylor GW, Fergus R. Adaptive deconvolutional networks for mid and high level feature learning. In: Proc of the 13th IEEE international conference on computer vision, ICCV. Barcelona, Spain; 2011. p. 2018–25.
[63] Kim H, Mnih A. Disentangling by factorising. In: Proc of the 25th international conference on machine learning, ICML. Stockholm, Sweden; 2018. p. 4153–71.
[64] Hoffman MD, Johnson MJ. ELBO surgery: yet another way to carve up the variational evidence lower bound. In: Proc of the 30th neural information processing systems, NIPS. Barcelona, Spain; 2016. p. 1177–83.
[65] Hjelm RD, Jacob AP, Che T, Trischler A, Cho K, Bengio Y. Boundary-seeking generative adversarial networks. In: Proc of the 6th international conference on learning representations, ICLR. Vancouver, Canada; 2018. p. 1–17.
[66] Zhao J, Mathieu M, LeCun Y. Metric-based generative adversarial network. In: Proc of the 5th international conference on learning representations, ICLR. Toulon, France; 2017. p. 672–80.
[67] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. In: Proc of the 34th international conference on machine learning, ICML. Sydney, Australia; 2017. p. 214–23.
[68] Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier GANs. In: Proc of the 34th international conference on machine learning, ICML. Sydney, Australia; 2017. p. 2642–51.
[69] Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In: Proc of the 30th conference on Neural Information Processing Systems, NIPS. Barcelona, Spain; 2016. p. 2180–8.
[70] Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA. Generative adversarial networks: an overview. IEEE Signal Process Mag 2018;35(1):53–65.
[71] Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T. Analyzing and improving the image quality of StyleGAN. In: Proc of the IEEE conference on computer vision and pattern recognition CVPR. Seattle, WA, USA; 2020. p. 8107–16.
[72] Durugkar IP, Gemp I, Mahadevan S. Generative multi-adversarial networks. In: Proc of the 5th international conference on learning representations, ICLR. Toulon, France; 2017. p. 25–45.
[73] Doan T, Monteiro J, Albuquerque I, Mazoure B, Durand A, Pineau J, Hjelm RD. On-line adaptative curriculum learning for gans. In: Proc of the 33rd AAAI conference on artificial intelligence. Honolulu, Hawaii, USA, vol. 33; 2019. p. 3470–7.
[74] Mukherjee S, Asnani H, Lin E, Kannan S. Clustergan: Latent space clustering in generative adversarial networks. In: Proc of the 33rd AAAI conference on artificial intelligence. Honolulu, Hawaii, USA; 2019. p. 4610–17.
[75] Miyato T, Koyama M. Cgans with projection discriminator. In: Proc of the 6th international conference on learning representations, ICLR. Vancouver, Canada; 2018. p. 1258–65.
[76] Huang X, Belongie S. Arbitrary style transfer in real-time with adaptive instance normalisation. In: Proc of the 19th IEEE international conference on computer vision, ICCV. Venice, Italy; 2017. p. 1501–10.
[77] Clevert D, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (ELUs). In: Proc of the 4th international conference on learning representations, ICLR. San Juan, Puerto Rico; 2016. p. 2569–78.
[78] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proc of the 10th international conference on artificial intelligence and statistics, ICAISC. Sardinia, Italy; 2010. p. 249–56.
[79] Hinton GE, Dayan P, Frey BJ, Neal RM. The“ wake-sleep” algorithm for unsupervised neural networks. Science 1995;268:1158–61.
[80] Kingma DP, Ba JL. Adam: A method for stochastic optimization. In: Proc of the 3rd international conference on learning representations, ICLR. San Diego, CA, USA; 2015. p. 1–15.
[81] Dowson D, Landau B. The fréchet distance between multivariate normal distributions. J Multivariate Anal 1982;12 (3):450–5.
[82] Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Proc of the 31st conference on neural information processing systems, NIPS. Long Beach, USA; 2017. p. 6626–37.
[83] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proc of the 3rd international conference on learning representations, ICLR. San Diego, CA, USA; 2015. p. 310–18.
[84] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proc of the IEEE conference on computer vision and pattern recognition, CVPR. Las Vegas, USA; 2016. p. 770–8.
[85] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. Imagenet large scale visual recognition challenge. Int J Comput Vis 2015;115(3):211–52.

[86] Mathieu MF, Zhao JJ, Zhao J, Ramesh A, Sprechmann P, LeCun Y. Disentangling factors of variation in deep representation using adversarial training. In: Proc of the 30th international conference on neural information processing systems, NIPS. Red Hook, NY, USA; 2016. p. 5047–55.

[87] Hameed K, Chai D, Rassau A. Class distribution-aware adaptive margins and cluster embedding for classification of fruit and vegetables at supermarket self-checkouts. Neurocomputing 2021;461:292–309.