Generalisation Techniques Using a Variational CEAE for Classifying Manuka Honey Quality

Tessa Phillips * and Waleed Abdulla†
Electrical, Computer and Software Engineering,
University of Auckland, Auckland, New Zealand
E-mail: *tphi709@aucklanduni.ac.nz, †w.abdulla@auckland.ac.nz

***Abstract*—This paper presents an integrated architecture of the class embodiment autoencoder (CEAE) and variational autoencoder. The aim is to improve the generalisation of the algorithm and accordingly increase the classification accuracy of unseen samples. The proposed variational CEAE is trained by using hyperspectral images of Manuka honey dataset, then evaluated for generalisation performance on unseen brands of honey. We applied well-known generalisation techniques to this structure, and evaluated the effect of these on our dataset. Our experiment results show that the average validation set performance of the new autoencoder technique on unseen brands is 55.4%, while the average benchmark technique is 48.1% for the same unseen brands. The autoencoder structures are performing feature reduction on our data, which has shown to improve the classification accuracy and generalisation performance. We tested the feature reduction techniques in combination with K-nearest-neighbour classifier, linear support vector machine (SVM), and radial basis function SVM. This work develops an important step toward the automatic classification of Manuka honey quality using hyperspectral imaging and machine learning. This is the first work to evaluate generalisation performance in honey classification, which is crucial for a viable real-world solution.**

***Index Terms*—Autoencoders, Feature Reduction, Generalisation, Hyperspectral Imaging, Support Vector Machines**

I. Introduction

This paper introduces the next step towards automating Manuka honey quality classification, using hyperspectral imaging and machine learning. The current real-world techniques are chemical-based, and are very accurate, but have several significant drawbacks. Honey is a large industry in New Zealand (NZ), some NZ honey such as Manuka honey, are very premium products making them a target for fraud.

Specifically, this paper aims to evaluate and combine existing generalisation techniques with a new type of autoencoder on our Manuka honey dataset. We are evaluating both their regular testing performance and the generalisation performance on unseen brands of honey. This generalisation is essential to simulate the real-world scenario, where honey of an unknown brand and type must be evaluated and given an accurate quality rating. This level of generalisation to unseen brands is already possible with the current chemical technique, as it analyses the honey for different concentrations of chemicals [1].

In our current work on honey classification, we have successfully been able to classify different botanical origins of honey from a range of different NZ honey brands [19]. We have also worked on more directly, the premium Manuka honey type. Initially by classifying the Unique Manuka Factor (UMF) for Manuka honey exclusively for each brand [16]. We have also classified Manuka honey types based on their Unique Manuka Factor (UMF) more generally across the entire dataset [20].

We split the dataset between training and testing so that every brand in the testing set is also in the training set. Every type of honey that we have has been sampled six times, with five of those samples in training and one for testing. This data split is useful because we can reliably train and test our algorithms on data from the same source [20].

In a real-world scenario identifying the UMF rating of honey, there might not be existing examples of the same type or brand of honey. This scenario means that our dataset and current work is so far not testing the generalisation ability to the standard required in the real-world case. Existing work on hyperspectral imaging for honey classification does not consider this real-world scenario, and use the same honey types in testing and training.

This paper uses a new experimental setup for Manuka honey classification, where we use a leave one out policy for each brand of Manuka honey. Each brand is excluded from the training set and used as a validation set. We then use the other brands for training and testing as usual. We focus on using different generalisation approaches to increase the performance on the validation set, while not compromising good performance on the testing set.

This evaluation approach was not possible using the existing Manuka UMF value dataset. Some UMF values only exist for one or two brands in the dataset, so they would not be a fair test of the ability to generalise. The new approach for this work changes the class labels into categories of UMF values. These categories are the ranges that are considered lower quality Manuka honey (UMF < 10), medium quality ($10 \leq$ UMF < 15), high quality ($15 \leq$ UMF < 20), and very high quality (UMF ≥ 20) [1]. These categories are all represented by many of the brands in the data. This approach makes it a fairer test when leaving out each of the brands.

We consider widely used generalisation techniques for machine learning algorithms, including regularisation [4], feature reduction, using simpler architectures, drop-out [26], and adding noise to the training data [8]. These are all strategies that have shown to help a learned model classify unseen data.

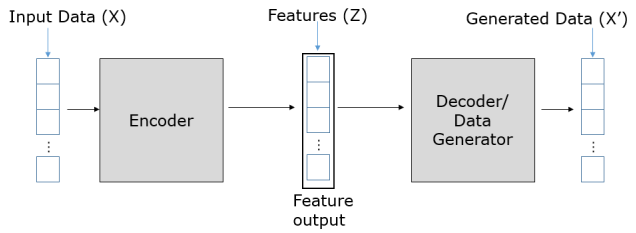

Fig. 1. Block diagram of the standard autoencoder architecture.

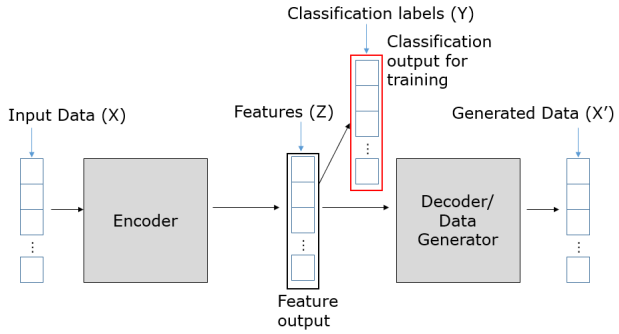

Fig. 2. Block diagram of the CEAE architecture.

However, these strategies are not typically used to classify data from a different source [7].

We also evaluate transfer learning inspired approaches for this work, alongside our existing generalisation techniques [27]. The honey classification problem requires transferring the knowledge from training with the primary dataset, to classify a new unseen brand. The honey spectrum changes due to many other attributes aside from the UMF rating, so generalisation of these brands is a hard problem. The data is expensive to acquire and time-consuming to capture; therefore, we look for techniques that can increase the generalisation without adding any new data.

We apply these generalisation techniques to the class embodiment autoencoder (CEAE), which previously performed the best in the wider honey dataset [19], [20].

The two main objectives of this work are:

- Evaluating honey classification using hyperspectral imaging in a way which better represents the real-world scenario.
- Creating a new autoencoder architecture to improve on the generalisation performance of the CEAE for this honey classification task.

II. Background

This section details the critical background information for this work. In particular, the feature reduction and classification techniques and existing work in honey quality classification.

A. Autoencoders

Autoencoders are a type of neural network which have two networks; encoder and decoder. The input data is passed through the encoder network to create features; then the features are passed through the decoder recreating the input data [2]. Figure 1 shows the general structure of an autoencoder. The number of features is typically smaller than the input dimension, and the autoencoder learns how to compress the input information.

1) Class Embodiment Autoencoder (CEAE): The CEAE was developed for this honey dataset [19] but is also applicable to general problems. The key part of this structure is the classification layer, which is weighted during training between zero and one. Figure 2 shows the structure for this autoencoder. We use the CEAE as a starting point for the new methods in this paper, as well as a key benchmark method for test set performance. The key differences between the CEAE architecture and standard supervised autoencoders is that a value weights the classification layer between zero and one in the loss function. The final output of the CEAE is the features, rather than the classification output.

2) Variational Autoenoders (VAEs): VAEs build upon the standard autoencoder technique, but instead of learning a single feature vector, they learn a distribution in the latent (feature) space for every training example. Figure 3 shows the VAE structure [25].

VAEs create new data by reconstructing the sampled latent space with the decoder network. The latent space created by VAEs satisfies certain conditions which make it easy to manipulate to create variations on the data. The latent space must be continuous and must map similar data close together in an organised way. Using regularisation on the latent space enforces these conditions [4].

This regularisation uses the Kullbeck-Leibler (KL) divergence between the returned latent distribution and a standard gaussian. The KL divergence refers to the relative entropy between the two distributions, equation 1 shows this regularisation term [4], [25]. Where D_{KL} is the KL divergence, x_i is a training example, z is a point in the latent space, $p(z|x_i)$ is the probability of a point z in the latent space, given an input x_i, and $q(z|x_i)$ is the estimated probability of a point z in the latent space given x_i.

$$D_{KL}(q(z|x_i)||p(z|x_i)) = \int q(z|x_i) log(\frac{q(z|x_i)}{p(z|x_i)}) dz \quad (1)$$

The regularisation that occurs involves substituting our distribution into equation 1. This substitution gives the final regularisation procedure in equation 2.

$$D_{KL}(q(z|x_i)||p(z|x_i)) = \frac{1}{2}[1 + log(\sigma_q^2) - \sigma_q^2 - \mu_q^2] \quad (2)$$

The final loss function at the decoder is the combination of the normal autoencoder loss and this regularisation, given in equation 3, where J is the dimension of the latent vector z, and L is the number of samples from the latent space. $E_{q(z|z_i)}$

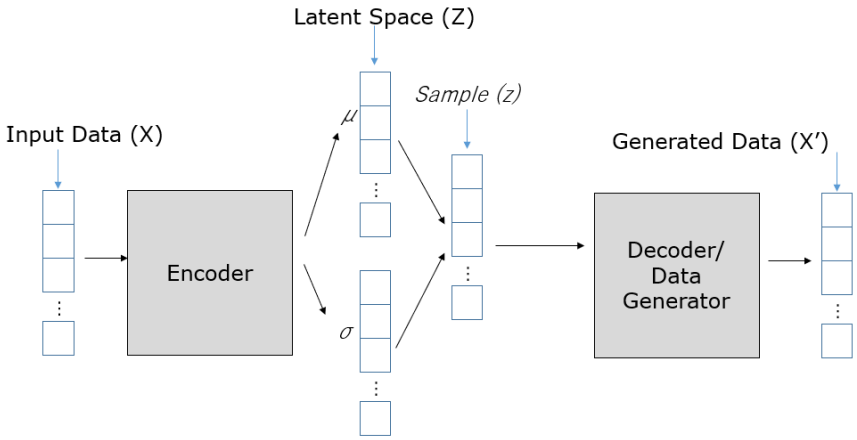

Fig. 3. Block diagram of the traditional VAE.

refers to the expectation of the following function with respect to the random variable $q(z|x_i)$ [4].

$$L = -\sum_{j=1}^{J} \frac{1}{2}[1 + log(\sigma_j^2) - \sigma_j^2 - \mu_j^2] - \frac{1}{L}\sum_{l=1}^{L} E_{\,q(z|x_i)}[log(p(x_i|z^{(i,l)}))] \quad (3)$$

By including this regularisation term in the loss function for the VAE, we can now manipulate the latent space. The VAE should also be able to generalise better to unseen data, as regularisation is a common technique for achieving this [4].

3) Denoising Autoencoders: Denoising autoencoders have been used widely for image applications, typically for removing noise in preprocessing [31]. The technique used in a denoising autoencoder is adding a random noise vector to each training example [32]. This additional noise means that the autoencoder learns to replicate the clean image at the output when given a noisy input image [8], [12].

B. Support Vector Machines (SVMs)

SVMs are well known for excellent generalisation performance, particularly in situations where the data is of high dimensionality, or has non-linear complexities [9], [10], [28], [30]. SVMs work by first transforming the feature space by a kernel function, and then splitting the data by a hyperplane in a way that minimises the risk of misclassifications. Figure 4 shows how an SVM can solve an example non-linear problem.

Previous work has investigated different SVM structures for Manuka honey classification [19] with varied success. We will consider these structures further in these generalisation experiments, as SVMs are known for good generalisation ability.

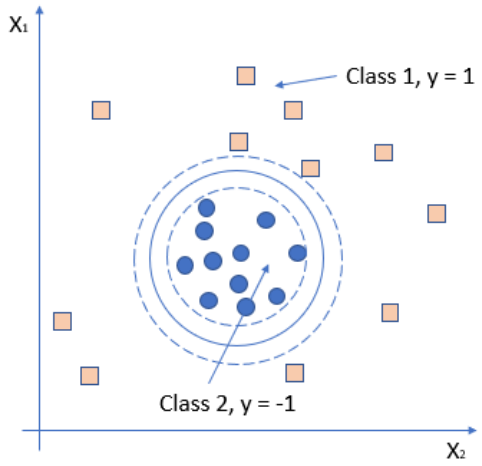

Fig. 4. An example non-linear problem being solved by an SVM.

C. Transfer learning

Transfer learning is the concept of taking the information learnt in one space and transferring that knowledge into another space [18], [29], [33]. Often this is done with similar spaces, but one might have more labelled information than the other. The general concept can apply to many situations where learning something in one space can help in learning something in another space [27].

Because we use deep autoencoders to learn our features, we consider transfer learning techniques from deep neural networks. These deep transfer learning techniques are separated into four categories [27]: instance-based, mapping-based, network-based, and adversarial-based. For our intended autoencoder algorithm, we focus on using mapping-based, instances-based, and network-based deep transfer learning techniques.

D. Manuka honey classification

Hyperspectral imaging has recently become a popular technique in analysing food quality. Hyperspectral imaging uses a combination of imaging and spectroscopy techniques. Hyperspectral image data is a 3D hypercube which contains both spatial and spectral information. Advantages of hyperspectral imaging over traditional methods for determining food quality are that it requires minimal sample preparation, fast data acquisition, and is non-destructive to the sample [5], [6], [34].

Spectroscopy has been used to determine the botanical origins of honey [3], [11], [17], [21], but more recently, Hyperspectral imaging is being investigated for this application.

Traditional cameras have been investigated for honey botanical origins classification, sometimes providing useful information in combination with other attributes [21]. Machine vision has been used to determine several attributes of food quality, such as colour, shape, and size. Using machine vision for classifying honey has been investigated [23]. Colour has been used as an important attribute in combination with machine learning to predict measurements of chemical properties such as ash content in different kinds of honey [22]. The dataset was minimal, and only used six different types of honey, with only 129 samples in total.

Hyperspectral imaging for honey classification is still very new, and there are only a few examples of this. There is one example of using hyperspectral imaging to detect if a pixel is sugar or honey in a mixture of sugar and honey [24]. The study was small, and only 56 honey samples were used.

A system for botanical origins of honey classification using hyperspectral imaging has been developed with a dataset of NZ honey [14]–[16]. Traditional machine learning approaches have been tested on this dataset, which resulted in an average balanced accuracy of 80.5% across all classes. Using the same dataset, a new type of autoencoder technique, the CEAE, was developed for feature reduction and applied to the honey classification scenario. The overall classification accuracy achieved over the entire dataset was 90% in combination with a k nearest neighbour (KNN) classifier [19].

Another study [13] has developed a technique using hyperspectral imaging to classify the botanical origins of five different kinds of honey, achieving 90% testing accuracy. However, this study used a minimal sample size, with only 52 samples in total, and only 20 of those samples were used for testing.

Overall, there has been minimal work on honey botanical origins classification using hyperspectral imaging, although there is now an extensive database developed for this application [20]. There has so far not been a focus on Manuka honey, nor generalisation for classifying unseen brands or honey types.

III. Methodologies

This section describes the methods we have developed intending to improve generalisation performance on the Manuka honey classification problem. The techniques we use are, for the most part, not new methods, but we are using new combinations of different techniques to form new architectures for a valuable application.

A. Variational Class Embodiment Autoencoder

We combine the benefits of the variational autoencoder and the class embodiment autoencoder to create the variational class embodiment autoencoder (VCEAE). We train this structure with three objectives: recreating the data, regularisation of the latent space, and classification accuracy.

These three objectives are all trained simultaneously, and the overall optimisation criteria is a combination of these three metrics. Figure 5 shows the VCEAE structure that we use in this work. The concept of including an additional classifier has been used previously with supervised variational autoencoders. However, the additional parameter weighting the classification output, and using the features as the output makes the VCEAE unique to the best of our knowledge.

Using the VCEAE architecture as a starting point, we then apply techniques that have shown to improve generalisation performance in other similar architectures. The VCEAE architecture Combines the CEAE which was the best technique on this dataset previously [19], and the VAE to improve generalisation.

B. Denoising VCEAE

Denoising autoencoders are powerful when working with noisy data. If we consider that some of our data could be quite noisy, using additional noise in training could improve classification and generalisation performance [8], [12], [31], [32].

C. Drop-out

Drop-out is a common technique used in deep learning to counteract overfitting of neural networks that can happen when training large models [26]. Drop-out works by temporarily removing some random nodes from the training process at each epoch. Including drop-out trains the network to have many redundant paths to the correct solution, and counters overfitting. The network will be more robust to slight changes at every network layer, including the input layer.

D. Data Transformation

We develop a data transformation approach for this data, using a mapping technique from deep transfer learning [27]. We are transforming the new brand, so it has the same distribution as the training set. We perform this in the latent space because the latent space has properties that allow it to be transformed based on the VAE.

We transform the data in the latent space of the learned VCEAE using a standard scalar approach. The distributions were scaled to match between the brand validation set and the training data. Based on initial testing, this scaling approach was more effective than transforming the data based on only the mean.

One drawback with this technique is that on data other than Manuka honey, this could skew the data and cause a

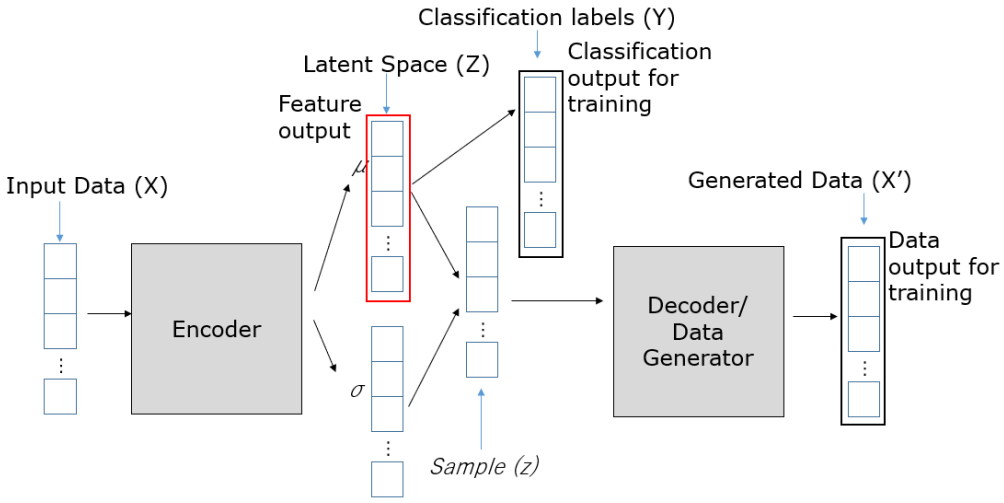

Fig. 5. Block digram of the VCEAE architecture. The classification output is connected to the latent space, and the output of the network is features from the latent space.

false positive for Manuka. False positives are undesired in our work, as it would mislabel a cheap kind of honey as the premium brand. One way to avoid this is to also develop a binary classifier for Manuka and non-Manuka before any transformations happen. The binary Manuka problem is much easier to solve, and does not have as much difficulty with generalisation.

E. Unsuperivsed training

We develop a new approach to training the network, inspired by instance-based deep transfer learning. In this case, we train the VCEAE as usual, but at the end apply a small number of unsupervised epochs to the VAE. This unsupervised training uses a combination of the brand validation data and the training set.

The idea of this approach is to get the variational autoencoder to regularise the latent space for the new unseen brand, as well as learning to compress the data from the unseen brand.

Another technique applying the same idea is to do this unsupervised training with only the unseen brand validation data. The idea is that this should allow the VAE to adapt to this specific dataset.

IV. Experiment Design

This section describes the design of the experiments. Particularly, the methods we use as benchmarks, the data split between training, testing, and validation, as well as the parameter tuning approach we take to obtain the network architecture.

A. Generalisation data splits

We are defining a new way to evaluate generalisation for our honey dataset that is more in line with the requirements of a real-world quality evaluation system.

Because our dataset is limited, it was not possible to test the generalisation extensively on the Manuka UMF value dataset. Instead, we created new data labels that put the Manuka honey samples into four categories that define the quality of the Manuka honey. These categories are in the ranges that are considered lower quality Manuka honey (UMF < 10), medium quality ($10 \leq$ UMF < 15), high quality ($15 \leq$ UMF < 20), and very high quality (UMF ≥ 20) [1]. These categories are all represented by many of the brands, which makes excluding each brand a fairer test. These tests are a step towards the final goal, which is being able to determine the exact UMF value of a Manuka honey sample regardless of the brand or type.

The way we evaluate generalisation is by leaving each brand out of the testing and training set and using the examples belonging to that brand as a validation set. The average performance on the validation set indicates how well the algorithm can classify honey from new or previously unseen brands. The new methods we have developed are aiming to improve the classification accuracy on this validation set.

B. Benchmark Methods

The benchmark methods we use have all been used for classification on this honey dataset in previous work. The first benchmark to consider is the original features from the hyperspectral image. The other benchmarks are principal component analysis (PCA), a standard autoencoder, a VAE, and a CEAE. The CEAE achieved the best performance on the wider honey dataset previously [19] but did not have the best generalisation ability.

The three autoencoder benchmarks will use the same layer structure decided for the VCEAE, but the number of epochs is tuned for each autoencoder. The PCA approach will take the first 20 principal components to keep in line with the autoencoders having an encoding dimension of 20.

KNN and two SVM methods (linear, and RBF kernel) will be applied after the feature reduction techniques to obtain

the final classification result. These final classifiers are the same as for the new generalisation approaches so that we can evaluate the feature reduction on different classifiers. Some initial tuning and evaluation will be done on only the KNN classifier, as the SVM classifiers require parameter tuning.

C. VCEAE Network Parameters

We developed the specific VCEAE architecture using a parameter sweep over the number of layers, the size of the latent feature space, and the classification weight. The best network found by the sum of test, training, and brand sets accuracy was selected to be our candidate for evaluation. This network was structured to have layer sizes that decrease evenly between the input feature size and the latent space feature size. The batch size of 32 and a learning rate of 0.001 are used in parameter tuning and also the initial evaluation. The number of epochs is later fine-tuned for the final architecture, but was set to number of layers $*$ 100 in this initial parameter tuning.

Overall the best structure found through our parameter tuning was a six hidden layer encoder and decoder network with layers $[128(input), 128, 110, 92, 74, 56, 38, 20(output)]$, and the reverse for the decoder. There are 20 features in the latent space, and it has a classification weight of 0.4. To find this network, we used 600 epochs with a batch size of 32, and a learning rate of 0.001 as defined in our parameter tuning process. The rectified linear unit (ReLU) activation function was used throughout the entire network, aside from the classification output, which used sigmoid. The number of epochs is fine-tuned between 0 and 500, evaluating every 50 epochs for each of the final autoencoder structures.

With our finalised network architecture, we then began tuning parameters for the generalisation techniques as well as performing some analysis on how the network performs with different values of these parameters. Section V shows this evaluation for our generalisation techniques that have some key parameters.

For SVMs, we tune the parameters C, and gamma (for RBF SVMs), using seven values on a log scale between 10^{-3} and 10^{3}, taking the best average result as our chosen parameters.

V. Results and Analysis

This section details the results of our generalisation experiments on the VCEAE architecture. The specific VCEAE structure we use is defined in section IV-C. We apply several techniques aiming to improve the generalisation ability of the network. As described in section IV-A, we evaluate performance on an unseen brand.

The first approach we apply is using drop-out. We add drop-out layers between every layer of our network with a low drop-out rate and evaluate how this can potentially improve the generalisation performance. Figure 6 shows how different drop-out rates affect the training, testing, and validation performance on our dataset.

We also consider using a minimal number of features in the latent space. This change involves leaving the network structure as it is but reducing the size of the feature layer.

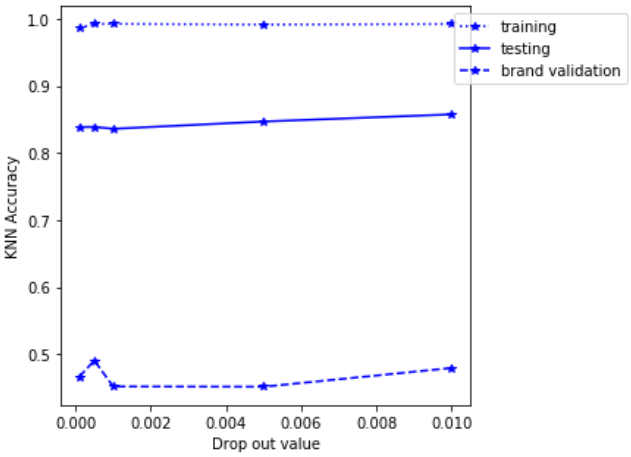

Fig. 6. Drop-out parameter tuning graph

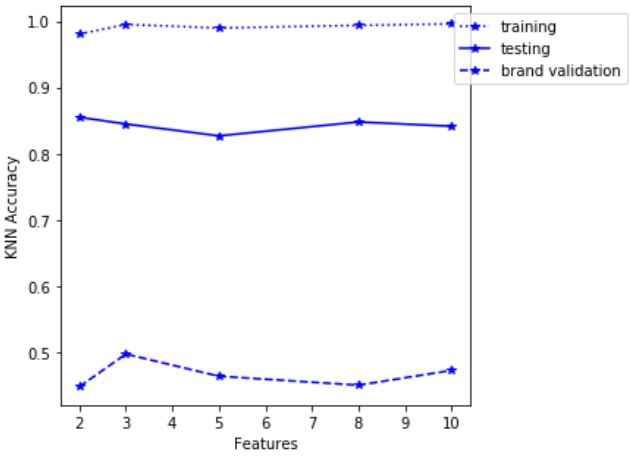

Fig. 7. Reduced features parameter tuning graph

Figure 7 shows how different feature sizes affect the overall performance of the architecture.

We also consider adding unsupervised training with the validation set at the end of our autoencoder training. Figure 8 shows the result of using a different number of training epochs to train the validation set unsupervised after regular network training. Figure 9 shows the result of unsupervised training the autoencoder with the validation set and the training set combined for a range of epochs after regular training of the network.

We also use a data transformation approach and a denoising autoencoder, as discussed in section III. These approaches do not require any additional parameter tuning.

Table V shows the results of these techniques. A KNN classifier with $K = 5$ is used after feature reduction on all the techniques, as it is a simple classifier that does not require any parameter tuning.

Table V shows that the VCEAE with reduced features technique is performing the best on the validation set, followed

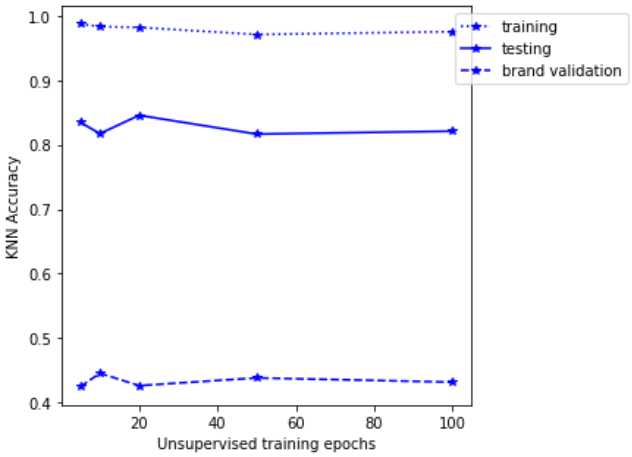

Fig. 8. Unsupervised training with validation set parameter tuning graph

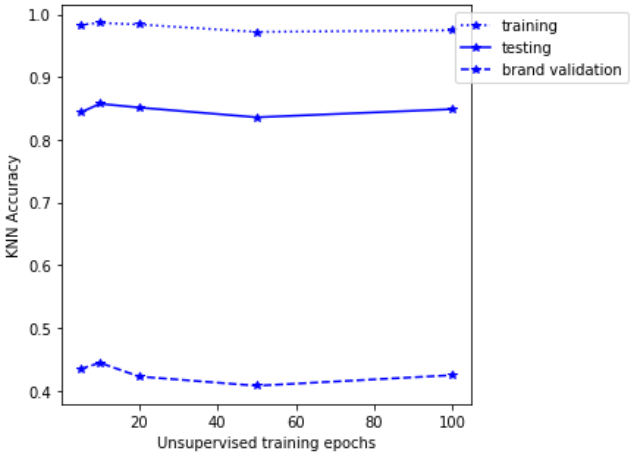

Fig. 9. Unsupervised training with validation + training set parameter tuning graph

closely by VCEAE with drop-out, and then the standard VCEAE. The other techniques did not have a positive effect on the validation data, which is surprising. With some further investigation, it can be theorised that the techniques are not suited to this type of generalisation problem. The denoising autoencoder did not improve the performance, which is likely because the variational autoencoder already includes random noise into the latent space during training. The other techniques were transfer learning based techniques. The idea behind this was to transfer the model and information learned in the training process to the unseen brand of honey. However, these data sets are not entirely different and share much information. These transfer learning techniques were possibly over-correcting on the new data and losing some useful classification information. Although transfer learning is a particularly exciting area to explore, these kinds of techniques are not suitable for our Manuka honey generalisation problem.

We progress using the two techniques (drop out and reduced features) that improved upon the standard VCEAE, and also combining the two techniques. These techniques are evaluated using SVMs as well as the KNN classifier. We consider both linear and radial basis function (RBF) kernel SVMs and compare the new techniques and the VCEAE to the existing benchmark techniques. Table V shows these techniques with SVMs applied, the combination of the two techniques, and the benchmark methods as described in section IV-B.

Comparing the results of the new methods in table V to the benchmark methods, we can see that when using the RBF SVM, our new techniques have an improvement over the known benchmarks in validation set performance. This

TABLE I
INITIAL RESULTS OF GENERALISATION TECHNIQUES BEING APPLIED TO THE VCEAE. BOLD INDICATES THAT THE BRAND VALIDATION PERFORMANCE WAS BETTER THAN THE STANDARD VCEAE.

Technique	Brand	Train	Test	Val
Standard VCEAE	C2	0.999	0.825	0.453
	C3	0.990	0.804	0.638
	C4	0.995	0.802	0.457
	C9	0.997	0.787	0.464
	C10	0.995	0.882	0.412
	C11	0.996	0.933	0.325
	avg	0.996	0.839	0.458
Denoising	C2	0.938	0.779	0.478
	C3	0.959	0.787	0.558
	C4	0.946	0.768	0.380
	C9	0.952	0.728	0.620
	C10	0.968	0.831	0.422
	C11	0.969	0.869	0.255
	avg	0.955	0.794	0.452
Drop-out	C2	0.996	0.811	0.371
	C3	0.995	0.792	0.658
	C4	0.993	0.788	0.500
	C9	0.995	0.829	0.676
	C10	0.995	0.882	0.433
	C11	0.983	0.933	0.305
	avg	**0.992**	**0.839**	**0.490**
Reduced Features (3)	C2	0.997	0.785	0.507
	C3	0.993	0.800	0.649
	C4	0.998	0.794	0.490
	C9	0.997	0.817	0.591
	C10	0.996	0.907	0.410
	C11	0.986	0.925	0.307
	avg	**0.995**	**0.838**	**0.492**
Data Transform	C2	0.991	0.777	0.396
	C3	0.983	0.829	0.638
	C4	0.982	0.774	0.240
	C9	0.990	0.783	0.564
	C10	0.989	0.844	0.417
	C11	0.990	0.933	0.311
	avg	0.987	0.824	0.428
Unsupervised val training	C2	0.975	0.766	0.380
	C3	0.987	0.802	0.640
	C4	0.983	0.826	0.500
	C9	0.990	0.802	0.533
	C10	0.984	0.778	0.458
	C11	0.984	0.930	0.158
	avg	0.984	0.817	0.445
Unsupervised val+train set training	C2	0.977	0.823	0.276
	C3	0.979	0.861	0.667
	C4	0.988	0.822	0.477
	C9	0.992	0.819	0.518
	C10	0.991	0.887	0.460
	C11	0.993	0.933	0.270
	avg	0.987	0.857	0.444

TABLE II
RESULTS OF SVMS APPLIED TO THE BEST GENERALISATION TECHNIQUES AND A COMBINATION OF THE BEST TWO TECHNIQUES. BOLD INDICATES THE BEST PERFORMANCE FOR EACH CLASSIFIER.

		KNN			Linear SVM			RBF SVM		
Technique	Brand	Train	Test	Val	Train	Test	Val	Train	Test	Val
New Methods										
Standard VCEAE	C2	0.999	0.825	0.453	1.0	0.783	0.596	0.999	0.806	0.418
	C3	0.990	0.804	0.638	0.925	0.699	0.771	0.997	0.798	0.618
	C4	0.995	0.802	0.457	0.959	0.818	0.500	0.972	0.842	0.433
	C9	0.997	0.787	0.464	0.944	0.743	0.431	0.993	0.783	0.351
	C10	0.995	0.882	0.412	0.996	0.824	0.418	0.996	0.904	0.400
	C11	0.996	0.933	0.325	0.995	0.936	0.304	0.969	0.939	0.263
	avg	0.996	0.839	0.458	0.970	0.801	**0.503**	0.988	0.845	0.414
Dropout	C2	0.996	0.811	0.371	0.996	0.783	0.347	1.0	0.754	0.349
	C3	0.995	0.792	0.658	0.992	0.787	0.667	1.0	0.783	0.918
	C4	0.993	0.788	0.500	0.994	0.804	0.487	1.0	0.778	0.500
	C9	0.995	0.829	0.676	0.992	0.872	0.578	1.0	0.806	0.558
	C10	0.995	0.882	0.433	0.993	0.909	0.418	0.997	0.824	0.427
	C11	0.983	0.933	0.305	0.980	0.925	0.341	0.989	0.904	0.273
	avg	0.992	0.839	0.490	0.991	0.847	0.473	0.998	0.808	0.504
Reduced Features (3)	C2	0.997	0.785	0.507	0.999	0.802	0.556	0.990	0.773	0.500
	C3	0.993	0.800	0.649	0.991	0.785	0.651	0.992	0.733	0.920
	C4	0.998	0.794	0.490	0.992	0.794	0.480	0.978	0.800	0.433
	C9	0.997	0.817	0.591	0.996	0.838	0.469	0.992	0.779	0.429
	C10	0.996	0.907	0.410	0.990	0.887	0.408	0.992	0.884	0.413
	C11	0.986	0.925	0.307	0.979	0.936	0.338	0.944	0.939	0.352
	avg	0.995	0.838	**0.492**	0.991	0.840	0.484	0.981	0.818	0.508
Dropout + Reduced Features	C2	0.997	0.823	0.287	0.989	0.792	0.34	0.935	0.724	0.449
	C3	0.998	0.792	0.667	0.995	0.804	0.62	0.947	0.718	0.696
	C4	0.994	0.816	0.413	0.991	0.812	0.497	0.909	0.76	0.847
	C9	0.994	0.817	0.44	0.997	0.838	0.46	0.909	0.771	0.458
	C10	0.996	0.853	0.39	0.996	0.907	0.42	0.976	0.776	0.377
	C11	0.987	0.933	0.291	0.987	0.931	0.284	0.945	0.859	0.497
	avg	0.994	0.839	0.415	0.993	0.847	0.437	0.937	0.768	**0.554**
Benchmark Methods										
CEAE	C2	0.997	0.821	0.458	0.977	0.779	0.498	0.992	0.804	0.484
	C3	0.995	0.773	0.662	0.979	0.752	0.658	0.986	0.766	0.662
	C4	0.996	0.818	0.483	0.99	0.802	0.457	0.995	0.818	0.483
	C9	0.997	0.792	0.402	0.996	0.787	0.391	0.996	0.785	0.402
	C10	0.996	0.893	0.385	0.994	0.878	0.363	0.995	0.88	0.373
	C11	1.0	0.933	0.247	0.997	0.928	0.285	0.998	0.931	0.245
	avg	0.997	0.838	0.44	0.989	0.821	0.442	0.994	0.831	0.442
VAE	C2	0.965	0.88	0.247	0.589	0.573	0.469	1.0	0.484	0.329
	C3	0.597	0.404	0.3	0.315	0.309	0.278	1.0	0.316	0.667
	C4	0.564	0.392	0.327	0.262	0.34	0.287	1.0	0.354	0.48
	C9	0.609	0.364	0.271	0.29	0.263	0.442	0.827	0.309	0.249
	C10	0.594	0.442	0.21	0.222	0.222	0.25	1.0	0.38	0.258
	C11	0.963	0.928	0.246	0.851	0.776	0.248	1.0	0.435	0.285
	avg	0.715	0.568	0.267	0.421	0.414	0.329	0.971	0.38	0.378
Standard AE	C2	0.979	0.829	0.46	0.849	0.794	0.276	0.974	0.768	0.376
	C3	0.982	0.863	0.664	0.82	0.752	0.64	0.973	0.716	0.664
	C4	0.979	0.75	0.31	0.78	0.776	0.217	0.973	0.736	0.4
	C9	0.983	0.832	0.611	0.868	0.705	0.464	0.984	0.796	0.731
	C10	0.985	0.876	0.488	0.845	0.764	0.313	0.987	0.836	0.492
	C11	0.989	0.885	0.156	0.867	0.773	0.147	0.982	0.933	0.156
	avg	0.983	0.839	0.448	0.838	0.761	0.343	0.979	0.797	0.47
Full Features	C2	0.989	0.819	0.293	0.894	0.747	0.202	0.93	0.808	0.322
	C3	0.991	0.857	0.636	0.828	0.659	0.793	0.923	0.821	0.667
	C4	0.99	0.836	0.453	0.841	0.718	0.47	0.938	0.814	0.487
	C9	0.992	0.863	0.5	0.88	0.712	0.464	0.962	0.869	0.658
	C10	0.99	0.873	0.407	0.841	0.724	0.553	0.952	0.851	0.402
	C11	0.997	0.92	0.158	0.914	0.755	0.404	0.947	0.957	0.169
	avg	0.991	0.861	0.408	0.866	0.719	0.481	0.942	0.854	0.451
PCA	C2	0.843	0.613	0.269	0.705	0.533	0.236	1.0	0.368	0.333
	C3	0.834	0.573	0.453	0.569	0.453	0.358	1.0	0.316	0.667
	C4	0.835	0.608	0.48	0.594	0.498	0.587	1.0	0.35	0.5
	C9	0.856	0.615	0.544	0.608	0.493	0.578	1.0	0.368	0.333
	C10	0.853	0.656	0.37	0.648	0.549	0.092	1.0	0.389	0.25
	C11	0.903	0.728	0.27	0.67	0.616	0.317	1.0	0.4	0.286
	avg	0.854	0.632	0.398	0.632	0.523	0.361	1.0	0.365	0.395

TABLE III
VALIDATION SET CONFUSION MATRICES FOR THE REDUCED FEATURES AND DROP-OUT VCEAE USING AN RBF SVM.

Brand	Manuka UMF value				
		0-9	10-14	15-19	20+
C2	0-9	114	36	0	0
	10-14	55	88	7	0
	15-19	150	0	0	0
	20+	0	0	0	0
		0-9	10-14	15-19	20+
C3	0-9	72	75	3	0
	10-14	59	241	0	0
	15-20	0	0	0	0
	20+	0	0	0	0
		0-9	10-14	15-19	20+
C4	0-9	139	11	0	0
	10-14	35	115	0	0
	15-19	0	0	0	0
	20+	0	0	0	0
		0-9	10-14	15-19	20+
C9	0-9	58	6	0	86
	10-14	2	148	0	0
	15-19	132	18	0	0
	20+	0	0	0	0
		0-9	10-14	15-19	20+
C10	0-9	121	0	4	25
	10-14	12	81	57	0
	15-19	8	118	24	0
	20+	148	0	2	0
		0-9	10-14	15-19	20+
C11	0-9	149	0	0	1
	10-14	150	126	0	24
	15-19	104	105	0	91
	20+	52	0	1	247

result shows that the techniques are having a positive effect in generalising to the unknown brand data. We can see some brands are performing very well, above 65% in many of the techniques, specifically brands C3 and C4 for our best technique. These high performing brands are increasing the average accuracy, so it is important to investigate why this might be happening in the data.

The best benchmark technique was the original features with a linear SVM with 48.1% accuracy on the validation set. This benchmark is more accurate than some of our new approaches on this classifier. However, the test performance was only 71.9%, where the new methods were all higher than this.

Interestingly, the new approaches improve a lot with the RBF SVM classifier. For the VCEAE with drop-out method and the reduced feature VCEAE method, the linear SVM did not improve the performance compared to a KNN classifier. The linear SVM did improve the performance when these approaches are combined and for the standard VCEAE.

We also analyse the confusion matrices from the best classifier in table V to determine how our approaches are performing very well for some brands, but less well for others.

Table V shows the confusion matrices for the VCEAE with drop-out and reduced features using an RBF SVM. Looking at the brands that performed very well, C3 and C4, we can see that these brands only have examples from the first two classes. These two classes are the largest in our training set in terms of the number of examples, and the number of brands containing examples of them. Having more brands containing these classes provides variation in the dataset so that the algorithms can generalise better for these two larger classes. For the smaller brands the classifier predicts most of the examples correctly if they are from either UMF0-9 or UMF10-14. Classifying examples from the UMF20+ group is working well in brand C11, but not in C10. All brands that have examples in UMF15-19 are unable to classify these examples very accurately. The classifier may be overfitting to one or two brands in training for these smaller classes, as there are not many brands that contain these classes. These examples are also from a range of different UMF values (15, 18); in contrast, UMF0-9 examples only contain UMF5 examples from our dataset. Having multiple classes meant the classifier has even fewer examples from each UMF class from UMF15-20. These small brands and classes are an area where we could look to improve the generalisation performance in future work, either by adding more data or by using the transfer learning approaches in a more targetted way.

VI. CONCLUSION AND FUTURE WORK

We have developed a new autoencoder architecture with the variational class embodiment autoencoder (VCEAE). This architecture is a combination of the CEAE which was previously the best performing structure on this dataset, and the VAE which is known to improve generalisation performance. In this paper, we introduced a new evaluation strategy to test the generalisation to unseen samples from brands of Manuka honey. The new VCEAE architecture had the best generalisation performance when using a small number of features, and drop-out in combination with an RBF SVM classifier. The accuracy has been increased by over 7% while retaining a high accuracy on the standard testing set. Increasing the generalisation ability of our algorithms is a step towards a system for real-world honey quality classification. This is the first work on honey classification to evaluate the performance of unseen types or brands of honey, which is critical for the real-world application. Future work will include adding more data to the database and involve finding ways to create fake data to improve the generalisation for small size classes.

REFERENCES

[1] *The New Zealand honey phenomenon in the USA*. New Zealand Consulate-General Los Angeles, 2015.

[2] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 37–49, 2012.

[3] Jessie Bong, Kerry M Loomes, Ralf C Schlothauer, and Jonathan M Stephens. Fluorescence markers in some new zealand honeys. *Food chemistry*, 192:1006–1014, 2016.

[4] Jaehoon Cha, Kyeong Soo Kim, and Sanghyuk Lee. On the transformation of latent space in autoencoders. *arXiv preprint arXiv:1901.08479*, 2019.

[5] Gamal ElMasry and Da-Wen Sun. Principles of hyperspectral imaging technology. In *Hyperspectral imaging for food quality analysis and control*, pages 3–43. Elsevier, 2010.

[6] A.A. Gowen, C.P. O'Donnell, P.J. Cullen, G. Downey, and J.M. Frias. Hyperspectral imaging - an emerging process analytical tool for food quality and safety control. *Trends in Food Science & Technology*, 18(12):590 – 598, 2007.

[7] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[8] Youngmoon Jung, Younggwan Kim, Yeunju Choi, and Hoirin Kim. Joint learning using denoising variational autoencoders for voice activity detection. In *Interspeech*, pages 1210–1214, 2018.

[9] Vojislav Kecman. *Support vector machines basics*. School of Engineering, University of Auckland, 2004.

[10] Vojislav Kecman. Support vector machines–an introduction. In *Support vector machines: theory and applications*, pages 1–47. Springer, 2005.

[11] Chellakutty Kumaravelu and Aravamudhan Gopal. Detection and quantification of adulteration in honey through near infrared spectroscopy. *International Journal of Food Properties*, 18(9):1930–1935, 2015.

[12] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. Speech enhancement based on deep denoising autoencoder. In *Interspeech*, pages 436–440, 2013.

[13] Saeid Minaei, Sahameh Shafiee, Gerrit Polder, Nasrolah Moghadam-Charkari, Saskia van Ruth, Mohsen Barzegar, Javad Zahiri, Martin Alewijn, and Piotr M Kuś. Vis/nir imaging application for honey floral origin determination. *Infrared Physics & Technology*, 86:218–225, 2017.

[14] Ary Noviyanto and Waleed H Abdulla. Honey dataset standard using hyperspectral imaging for machine learning problems. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 473–477. IEEE, 2017.

[15] Ary Noviyanto and Waleed H Abdulla. Segmentation and calibration of hyperspectral imaging for honey analysis. *Computers and Electronics in Agriculture*, 159:129–139, 2019.

[16] Ary Noviyanto and Waleed H Abdulla. Honey botanical origin classification using hyperspectral imaging and machine learning. *Journal of Food Engineering*, 265:109684, 2020.

[17] Ary Noviyanto, Waleed Abdullah, Wei Yu, and Zoran Salcic. Research trends in optical spectrum for honey analysis. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific*, pages 416–425. IEEE, 2015.

[18] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[19] Tessa Phillips and WH Abdulla. Class embodiment autoencoder (ceae) for classifying the botanical origins of honey. In *Image and Vision Computing New Zealand (IVCNZ)*, 2019.

[20] Tessa Phillips, Ary Noviyanto, and Waleed Abdulla. Hyperspectral imaging honey database. 4 2020. *Available at https://figshare.com/s/25afe30ff531b8f1e65f*.

[21] Kaspar Ruoff. *Authentication of the botanical origin of honey*. PhD thesis, ETH Zurich, 2006.

[22] Sahameh Shafiee, Saeid Minaei, Nasrollah Moghaddam-Charkari, and Mohsen Barzegar. Honey characterization using computer vision system and artificial neural networks. *Food chemistry*, 159:143–150, 2014.

[23] Sahameh Shafiee, Saeid Minaei, Nasrollah Moghaddam-Charkari, Mahdi Ghasemi-Varnamkhasti, and Mohsen Barzegar. Potential application of machine vision to honey characterization. *Trends in food science & technology*, 30(2):174–177, 2013.

[24] Sahameh Shafiee, Gerrit Polder, Saeid Minaei, Nasrolah Moghadam-Charkari, Saskia van Ruth, and Piotr M. Ku. Detection of honey adulteration using hyperspectral imaging. *IFAC-PapersOnLine*, 49(16):311 – 314, 2016. 5th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2016.

[25] Irhum Shafkat. Intuitively understanding variational autoencoders.

[26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[27] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer, 2018.

[28] Yichuan Tang. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*, 2013.

[29] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[30] Vladimir Vapnik. The support vector method of function estimation. In *Nonlinear Modeling*, pages 55–85. Springer, 1998.

[31] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[32] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

[33] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):9, 2016.

[34] Di Wu and Da-Wen Sun. Advanced applications of hyperspectral imaging technology for food quality and safety analysis and assessment: A review-part i: Fundamentals. *Innovative Food Science & Emerging Technologies*, 19:1–14, 2013.