IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 February 2023, accepted 7 March 2023, date of publication 10 March 2023, date of current version 16 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3255795

RESEARCH ARTICLE

GMM-IL: Image Classification Using Incrementally Learnt, Independent Probabilistic Models for Small Sample Sizes

PENNY JOHNSTON, KEILLER NOGUEIRA, AND KEVIN SWINGLER
Department of Computer Science, University of Stirling, FK9 4LA Stirling, U.K.

Corresponding author: Penny Johnston (Penny.Johnston@stir.ac.uk)

This work was supported by Stirling University.

ABSTRACT When deep-learning classifiers try to learn new classes through supervised learning, they exhibit catastrophic forgetting issues. In this paper we propose the Gaussian Mixture Model - Incremental Learner (GMM-IL), a novel two-stage architecture that couples unsupervised visual feature learning with supervised probabilistic models to represent each class. The key novelty of GMM-IL is that each class is learnt independently of the other classes. New classes can be incrementally learnt using a small set of annotated images with no requirement to relearn data from existing classes. This enables the incremental addition of classes to a model, that can be indexed by visual features and reasoned over based on perception. Using Gaussian Mixture Models to represent the independent classes, we outperform a benchmark of an equivalent network with a Softmax head, obtaining increased accuracy for sample sizes smaller than 12 and increased weighted F1 score for 3 imbalanced class profiles in that sample range. This novel method enables new classes to be added to a system with only access to a few annotated images of the new class.

INDEX TERMS Image classification, incremental learning, probabilistic models, small sample sizes, deep learning.

I. INTRODUCTION

Incremental learning of new classes without forgetting old classes is essential for real-world problems but extremely challenging for modern deep learning methods. Current incremental deep learners suffer from 'catastrophic forgetting' when after learning Class A, they are then required to learn Class B. The issue occurs due to the sharing of a set number of weights in the neural network. These are optimised for Class A, however, in order to learn Class B the weights must be altered, resulting in new knowledge overwriting previous knowledge.

In order to overcome this catastrophic forgetting, we introduce a universal function approximator, in the form of independent Gaussian Mixture Models (GMMs). This enables a separation of task between; learning the principal visual features and learning the class definition. The GMMs also enable additional clustering and generative task functionality. Taking our inspiration from humans, and reflecting the identified tasks, a two stage pipeline is created where images are first encoded to visual features, which are then used during the modelling of independent classes. In stage one as shown in Figure 1, the visual features are learnt by an autoencoder using unlabelled images, this creates a latent space representation. Then in stage two each probabilistic model is independently trained to learn a class conditional probability distribution over that latent space. New classes can then be added without having to retrain the visual features or relearn previously learnt classes. The probabilistic model we use is a Gaussian Mixture Model. Using this architecture we can translate, label to image or image to label via the encoder and decoder created when training the Autoencoder. The independent class models can be enriched with symbolic information and stored in an extensible knowledge graph. This proposed neuro-symbolic architecture creates a specific structure at four levels and each level is aligned

The associate editor coordinating the review of this manuscript and approving it for publication was Jeon Gwanggil.

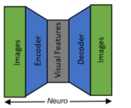
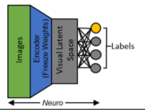
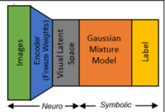
Training Pipeline Stage	STAGE 1	STAGE 2	
Model Name	Autoencoder	SoftMax (Benchmark)	GMM-IL
Model Architecture	 Images Encoder Visual Features Decoder Images Neuro	 Images Encoder (Freeze Weights) Visual Latent Space Labels Neuro	 Images Encoder (Freeze Weights) Visual Latent Space Gaussian Mixture Model Label Neuro Symbolic
Training Dataset	Dataset 1 Vast number of image pairs in high compute environment.	Dataset 2 Small number of images requiring small compute environment	
Training Type	Unsupervised	Supervised	Supervised
Training Sample Size	Large Quantity of Unlabelled Images	Medium Quantity of Labelled Images	Small Quantity of Labelled Images
Training Pairs	Image to Image (Pairs)	Image to Label (Pairs)	Image to Label & Label to Image
Training Purpose	Train Visual Features	Train model to discriminate between **ALL labels.**	Train model to discriminate **A Label**.
Learning Over Time	Need all previous image samples	Need all previous image samples	Needs only image samples for the new Label.
Bottleneck Dependency		Shared Network Weights	Shared Visual Features
Model can generate images.	Yes	No	Yes
Model can be indexed by a label	No	No	Yes
Can find a distance between Images	Yes	No	Yes

FIGURE 1. Two stage training process. Comparison of autoencoder, GMM-IL and benchmark model attributes.

with a human concept and supports open-ended learning which combines the strengths of the symbolic approaches with insights from machine learning. Figure 3 shows an overview of the proposed architecture, the analogies between humans and machines are shown below for each of the four levels:

1) Human: Learns to see visual information.
   Machine: Learns a latent space of visual features during Autoencoder training.
2) Human: After seeing an object, are taught a name to give it meaning.
   Machine: After creating prior visual features, train a probabilistic model to represent a class which gives that group of training images a symbolic meaning.
3) Human: Learn a new object without needing to see previous objects at the same time. Previously learnt objects can be imagined.
   Machine: Train a new class in the form of a probabilistic model without requiring access to previous training data. Previously learnt models can be sampled and the resulting latent embeddings decoded into images.
4) Human: Identify objects they are paying attention to in their field of view in real-time.
   Machine: Carry out object classification for the contents of an image at inference time.

This paper is organised as follows: In Section II we place our GMM-IL within the ontology of Incremental Learners with associated discussion. We detail the proposed GMM-IL in Section III, document our experimental setup in Section IV, and subsequently report our results in Section V. In Section VI we suggest possible improvements of this method. Finally, we conclude in Section VII and comment on possible future research directions.

A. CONTRIBUTIONS OF THIS PAPER

To the best of our knowledge, there is no similar GMM-Incremental Learner in the literature. The value of our proposed method is in the delivery of:

- A novel class representation, which couples transferred visual feature learning with independent probabilistic class learning and is easily extended to accommodate new classes.
- Gaussian Mixture Models, which can be trained using small sample sizes, decreasing both model training time and the need for costly annotated images.
- A novel classifier that exhibits no catastrophic forgetting issues due to the separation of the shared weights found in the fully connected and softmax layers of standard deep learning classifiers.

II. RELATED WORK

Incremental Learning aims at incrementally updating a trained model, through tasks that learn new classes without forgetting old classes [1], [2] [3]. Class Incremental Learning is where a limited memory or no previously learned samples are allowed during the training process. This limitation is motivated by practical applications, such as storage and computing constraints which prevent us from simply retraining the entire model for each new task. It is worth mentioning, that Incremental Learning is different from Transfer Learning in that it also aims to have good performance in both old and new tasks. Since the objective of an Incremental Learner is to keep on learning new tasks, it should be evaluated based on the classifier's performance on the past and the present tasks, in order to be confident about its behaviour in future unseen tasks. Lopez-Paz and Ranzato [4] pointed out that the ability of learners to transfer knowledge should also be paid attention to, and accordingly proposed the concepts of backward transfer (BWT, which is the influence that learning a task has on the performance of previous tasks) and forward transfer (FWT, which is the influence that learning a task has on the performance on future tasks). These new metrics are emerging which balance intransigence v forgetting [5].

A. INCREMENTAL LEARNING CHALLENGES

Catastrophic Forgetting (CF) identified by McCloskey and Cohen [6] over 30 years ago is when new learning interferes with the old learning, resulting in a reduced accuracy. Ideally, keeping the network's weights stable prevents previously learned tasks from being forgotten, but too much stability prevents the model from learning new tasks. The essence of the stability-plasticity dilemma describes how to design a balanced system that is simultaneously sensitive to but not radically disrupted by new inputs and therefore can incrementally learn [7].

B. ONTOLOGY OF INCREMENTAL LEARNERS

The standard Incremental Learner models use a neural network framework, which intrinsically creates several

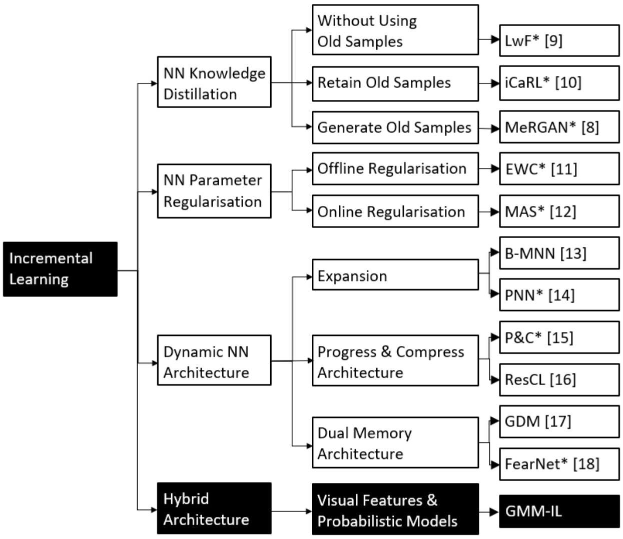


FIGURE 2. Ontology of Incremental Learners, placement of GMM-IL in categories taken and adapted from Liu et al [8]. The models shown are the most cited according to Liu et al (*) or we have mentioned it in the related work. Model References: LwF [9], Incremental classifier and representation learning (iCaRL) [10], Memory Replay GANs (MeRGAN) [8], Elastic Weight Consolidation (EWC) [11], Memory Aware Synapses: Learning What (not) to forget. (MAS) [12], Knowledge transfer in deep block-modular neural networks. (B-MNN) [13], Progressive Neural Networks (PNN) [14], Progress & Compress (P&C) [15], Residual continual learning. (ResCL) [16], Lifelong learning of spatiotemporal representations with dual-memory recurrent self-organisation. (GDM) [17], Fearnet: Brain Inspired Model for Incremental Learning. (FearNet) [18], GMM-IL: Gaussian Mixture Model Incremental Learner. (GMM-IL).

challenges such as; catastrophic forgetting, memory limitation and concept drift. The Ontology we adopt can be seen in Figure 2 which has been adapted from a Taxonomy by Lui et al [8] and selected due to its structural categories. Other useful surveys and ontologies related to this field are [1], [3], [19], [20] and [21]. We reflect where GMM-IL should be placed based on its structure within this context. The three current categories specified to overcome Incremental Learning issues are:(1) Parameter Regularisation; (2) Knowledge Distillation; and (3) Dynamic Architecture.

Parameter Regularisation based methods utilise regularisation techniques such as constraining the update of important parameters, dropout and early stopping. These are all aimed at retaining previous task knowledge. Knowledge distillation methods distil knowledge from an old model into the current model. The various ways this is carried out are by; (1) retaining old samples i.e. Incremental classifier and representation learning (iCaRL) [10], (2) without using old samples and (3) generating old samples. With the addition of new tasks, most dynamic architecture methods flexibly adjust the network structure.

In the GMM-IL method, visual feature knowledge is represented in a static latent space, and the symbolic knowledge does not depend on any shared weights, so it does not require parameter regularisation. Whilst visual information is distilled into the latent space, this only needs to be done once during classifier initialisation. There is no need to manipulate or save old training samples in the form of exemplars during the incremental training of tasks. For these reasons GMM-IL would belong in the Dynamic Architecture category.

Dynamic Architecture strategies include; (1) expansion, (2) progress and compress (P&C) and (3) dual memory (D-M) architectures. Black-Modular neural networks (B-MNN) [13], Progressive neural networks (PNN) [14] both augment the existing neural net with a 'piggy-back' neural net throughout the structure which gets trained on the new task. Not to be confused with fine tuning which adds one additional layer to a frozen memory. The main disadvantage of this approach is that the amount of parameters is exponentially proportional to the number of learned tasks. Progress and Compress (P&C) architectures maintain a constant number of parameters and consist of two parts, a knowledge base and the active column. The compression phase extracts the knowledge learned in the previous expansion phase to the knowledge base, and uses the Elastic Weight Consolidation (EWC) [11] strategy to protect the previously learned knowledge. In the expansion phase, the learning of new tasks reuses the characteristics of the knowledge base through lateral connections. The training approach alternates to limit expansion of the model while completing knowledge retention. These methods have limitations in scalability when it comes to multi-task incremental learning scenarios. Dual memory architectures are based on complementary learning systems (CLS) theory [22], [23]. The hippocampus system and the neocortex system balance the fast learning and slow learning processes. Therefore, the general dual memory architecture includes long and short-term memory. The former is used for memorising past learning experiences and the later for learning current tasks. Growing Dual-Memory (GDM) [17] considers the impact of continuous data over time on incremental learning.

Whilst the memory is expanded, each symbolic probabilistic model is small in size and expands at a rate of one model per learnt class. A compress and expand strategy is used, but it fulfils a very different function. The compress is for the visual features and the expand is in the form of building symbolic definitions. GMM-IL does have a dual memory in the form of visual features held separately to symbolic definitions. We justify the creation of a new category called, 'Hybrid Architecture' by the fact that our proposed architecture does not solely use a neural net. This is created for future Incremental Learners that capture the benefits of neural nets and combine them with other models such as the probabilistic models in GMM-IL. Following the trend within the ontology we add a structural sub category called, 'Visual Features & Probabilistic Models', this reflects our separate visual and symbolic structure that delivers system stability and flexibility. We name our classifier 'GMM-IL : GMM Incremental Learner', which identifies Gaussian Mixture Models to be the type of probabilistic model used.

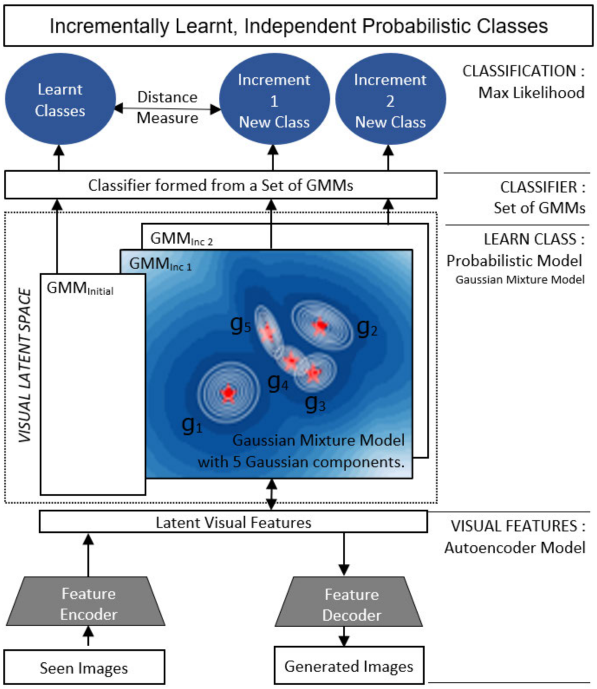


FIGURE 3. GMM-IL : Incrementally learnt, Independent probabilistic class models. (1) Autoencoder model,(2) Probabilistic model, (3) Classifier, (4) Classification logic.

III. METHOD

Our aim is to classify each visual concept using incremental learning, trained on small sample sizes using a hybrid architecture. The proposed architecture is modular in nature, enabling drop-in replacements for the autoencoder and probability models. A description of the selection and training of these models can be found in Section IV-E.

The four levels and their interactions are shown in Figure 3, from the bottom up they are:

1) An *Autoencoder Model* trained once on a large corpus of unlabelled data, enabling generalised useful visual features to be extracted from the image corpus. Detailed in Section III-A.
2) Independent *Probabilistic Models* which form the class definition, independently trained on a small number of visual features. Visual features are a result of encoding the associated class images. Detailed in Section III-B.
3) A *Classifier* comprised of a set of learnt Probabilistic Models which can be added to as new class data become available. Detailed in Section III-C.
4) *Classification* logic can be carried out across all the probabilistic models to evaluate the likelihood that at inference time, a specific image belongs to a class. Detailed in Section III-D.

A. AUTOENCODER

The Autoencoder transforms an image from a high dimensional space to a lower dimensional space for ease of manipulation. The main premise is that unsupervised training initialises this encoder on a vast number of image samples in a high compute environment. When encoding visual features, we are not only interested in the autoencoder's ability to reconstruct the input image, but also on encoding a useful representation. By useful we mean the representation is not task specific, is spread throughout the latent space and contains visual motifs at different scales. These attributes will enable it to generalise well to unseen symbolic classes. We selected a vanilla autoencoder since Chadebec and Vincent [24] carried out a case study benchmark, where they presented and compared 19 generative autoencoder models. They found that the autoencoder which did not try to manipulate the latent space in end-to-end training produced the highest classification accuracy.

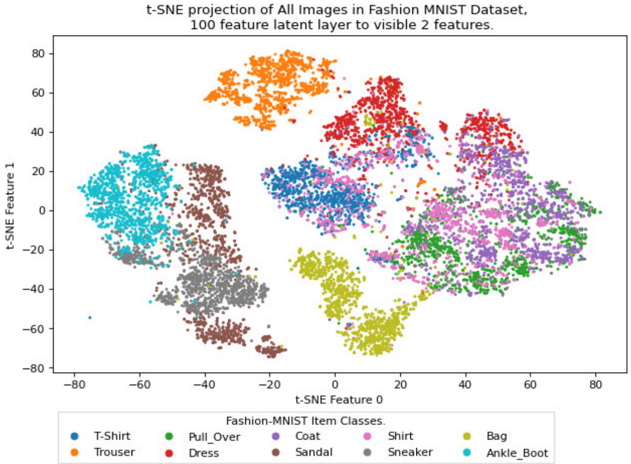


FIGURE 4. t-SNE Plot of Latent Space created through unsupervised Autoencoder training on the full Fashion-MNIST dataset, coloured by ground truth Class. Original feature number = 100.

To aid intuition in Figure 4 we have visualised the latent space for the Fashion-MNIST dataset. Each image was encoded to a visual embedding with 100 features and then using the t-SNE (t-distributed Stochastic Neighbour Embedding) method [25] has been projected to 2 visual features so that a 2D plot could be created. Note that no labels were involved in the training of our latent space. This results in a learnt structure of the most significant perceptual characteristics of images which is not biased by any symbolic labels. This enables the selected features to generalise well to future unseen labels. The plot also shows how well the training images have been split up in latent space purely based on their visual features and human selection of the images. Once an image has been encoded as visual features, the decoder can be used on those visual features to reconstruct the original image.

B. PROBABILISTIC MODEL

Once a visual feature latent space has been established, we can build specific concepts on top. We do this by selecting representative images for our concept, encoding those images into visual features using the encoder from the autoencoder and then training a probabilistic model using those encodings. The probabilistic model we selected is a Gaussian Mixture Model (GMM) [26] which is a universal function approximator, in that given a sufficient number of components, it can approximate any smooth function to arbitrary accuracy [27]. A Gaussian Mixture Model is a weighted sum of M component Gaussian densities as given by equation (1),

$$p(\boldsymbol{x}|\lambda) = \sum_{i=1}^{M} w_i g(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}) \tag{1}$$

where the notation used is; $\boldsymbol{\mu}_i$ for mean, $\boldsymbol{\Sigma_i}$ for covariance matrix and $\boldsymbol{\lambda}$ to express the collection of all component parameters $\boldsymbol{\lambda} = (w_i, \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$, which contains weights, means, and covariance matrix respectively for all Gaussian components. $\boldsymbol{x}$ is a D-dimension continuous-valued data vector (i.e. visual features), $w_i, \quad i = 1, \ldots, M$, are the mixture weights, and $g(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}), \quad i = 1, \ldots, M$ are the component Gaussian densities. Each component density is a D-variate Gaussian function of the form shown in equation (2),

$$g(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\boldsymbol{\Sigma_i}|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)} \tag{2}$$

with mean vector $\boldsymbol{\mu_i}$ and covariance matrix Σ_i. The mixture weights satisfy the constraint that $\sum_{i=1}^{M} w_i = 1$. The complete Gaussian mixture model is parameterised by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation,

$$\lambda = w_i, \boldsymbol{\mu}_i, \Sigma_i \qquad i = 1, \ldots, M. \tag{3}$$

There are several variants on the GMM shown in equation (3). The covariance matrices, Σ_i can be full rank or constrained to be diagonal. Additionally, parameters can be shared, or tied, among the Gaussian components. Each GMM is trained on a dataset that represents a concept. The Maximum Likelihood Estimation (MLE) for normal mixtures and Estimation Maximisation (EM) algorithm [28] are used after setting the training hyper-parameters. The GMM is initialised using K-Means centroids for the first Estimation step. Using a grid search we produce candidate GMMs from which we select the model with the lowest Bayesian Information Criterion (BIC) validation score (to prevent over-fitting) to represent that concept.

Each Class GMM is comprised of a number of components each with a mean and co-variance. The diagonal of the co-variance gives the variance of the component and the rest of the matrix describes the relationship between each of the features dependent on the component type. Figure 5 illustrates a reduced dimensional Gaussian Mixture Model with 2 components fitting bivariate distribution, with respective probability density distributions in shared axes for the Ankle Boot Class in the data.

C. CLASSIFIER

Once the probabilistic model for one class has been learnt, the next one can be incrementally learnt by simply training it and adding it to our set of GMMs in the classifier. This requires only the training data for the current class being learned and none of the training data for previous classes. Also, for the classifier to forget a class, it is as simple as removing the probabilistic class from the classifier set.

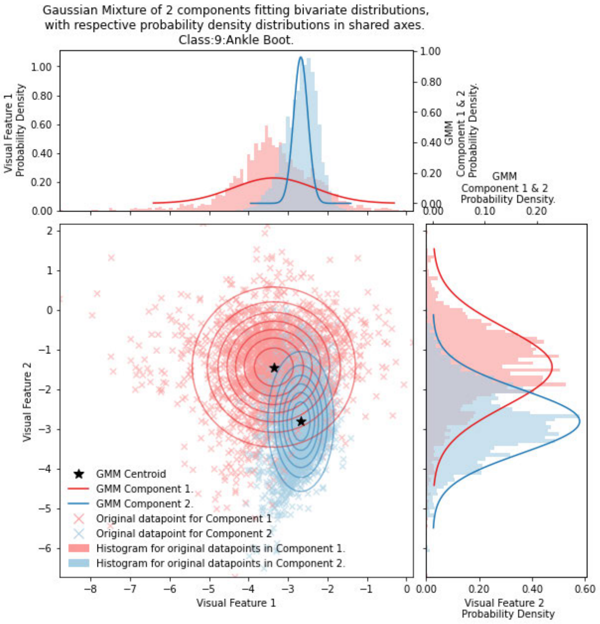


FIGURE 5. **Gaussian mixture model with 2 components fitting bivariate distribution, with respective probability density distributions in shared axes for Ankle Boot Class.**

In order to help with the intuition of a classifier comprised of a set of GMMs used for classification, we have built 10 GMMs based on a 2 feature encoder, this then enables us to create a 2D visualisation as shown in Figure 6. This map shows where individual GMM component distributions are, in the form of their GMM component mean values (stars) and co-variance contours (ellipses). We generate 2500 values for feature 1 & 2 which represent encoded images, they cover our space and generate a map of what the predicted classification will be at each of these points, based on a maximum likelihood score. Only a few classes are shown so that the means and co-variances are easier to see. Classes that are visually similar will have similar GMMs, leading to reduced discriminatory power.

D. CLASSIFICATION LOGIC

At inference time all the class likelihoods are evaluated for the GMMs in the classifier and the class with the maximum likelihood score is selected as the assigned classification. A pairwise GMM distance can be calculated which gives an indication of how similar the GMM representations are, and hence the extent of the classifiers discriminatory power. This was tested by creating a pairwise distance matrix for the Fashion-MNIST dataset, using the Jensen-Shannon method, then correlating it with the confusion matrix created at inference time. The resulting Spearman correlation coefficient was

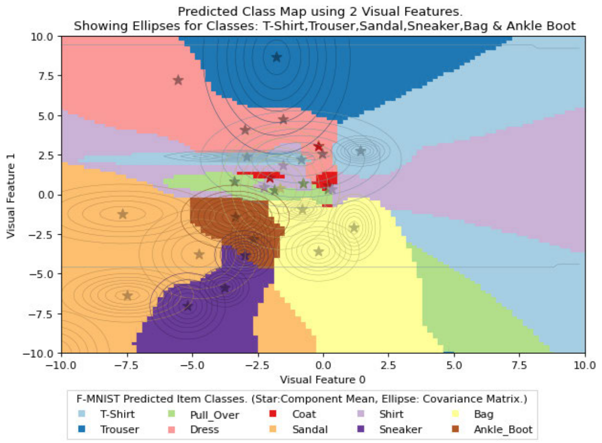


FIGURE 6. Map of the predicted classification for a 2 feature image encoder, overlaid with 7 GMMs and their component means (stars) and covariance contours (ellipses) on the Fashion-MNIST dataset.

0.78 with a p-value of 0, which indicated that the similarity matrix could predict the level of classifier errors to some extent.

IV. EXPERIMENT SETUP

A. HARDWARE AND SOFTWARE

All deep learning-based models were implemented using TensorFlow [29] Version 2.7.0. The code was written in Jupyter notebooks with Python Version 3.7.3. and CUDA version 11.2. All experiments conducted here were performed on a 64-bit Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz workstation with 64 CPU cores and 768GB RAM. NVIDIA® GeForce (driver version 495.44) with 4*GTX1080Ti each with 11GB RAM. Debian version 10.12 was used as the operating system. The scikit-learn and pycm [30] library were used for Metrics.

B. DATASETS

The performance of our model and the benchmark model are evaluated on the public dataset Fashion-MNIST [31] which contains gray scale images of 10 clothing categories. The official training dataset was split into, 80% creating a new training dataset (48K) and 20% creating a new validation dataset (12K). 100% of the official test dataset (10K) was used for our test set. Within each dataset all classes contained the same number of images, where this changes in our experiments it is noted in that experiments section. We use the same dataset for 2 purposes:

1) Dataset 1 - Autoencoder Model Training: To train the Autoencoder model to create Visual Features. The premise is that the Autoencoder model will learn through unsupervised training on the largest image dataset possible, using vast computer power in a big data paradigm. That once carried out the resulting encoder/decoder will then be used on all vision tasks without alteration (frozen weights). However, for the experiments in this paper the dataset above is used to investigate if the classifier can learn unseen classes without that class having being used during the training of the Autoencoder.
2) Dataset 2 - Probabilistic Model Training: To train the GMMs to create independent symbolic class representations. It is this dataset that is manipulated to investigate the impact on the classifier accuracy for; sample size, imbalanced classes and incrementally learnt class definitions.

C. EVALUATION METRICS

Quantitative metrics of accuracy score, weighted F1 score and Cohen Kappa are reported. The predictive accuracy metric measures the difference between the imputed values and their corresponding actual values. The weighted F1 score is used since class imbalance is investigated. Since multi class classification is carried out, Cohen's kappa is used to measure the agreement between GMMs which each classify N images into C mutually exclusive classes.

D. DATA CONSISTENCY

When an experiment contains a suite of increasing or decreasing sample sizes, a dataset is managed to contain the same images as previously used to ensure experimental consistency. All reported test results are carried out using 100% of the held out test dataset unless otherwise stated in an experiment.

E. MODEL SETUP

1) AUTOENCODER

The encoder has two convolution layers (followed by ReLU activations [32]) with 3 × 3 filters, applied with a stride of 2 and padding to maintain the same size image. From layer to layer, the number of filters (initially, 32) is doubled. The output of the last convolution layer is flattened and then mapped into a configurable dense layer which creates the features of our latent space. The decoder mirrors the encoder, using convolutional transpose operators [33]. The full architecture is shown in Figure 7.

The autoencoder uses Adam [34] for optimisation. The learning rate is reduced according to a cosine function [35]. The following hyper-parameters to define the search space were defined through several experiments, a base learning rate of 0.003, and a final learning rate of 0.001, a maximum number of 20 updates, 5 warm up steps and trained with 40 epochs. The batch size was set to 50. Training was carried out using unannotated images using a Mean Squared Error loss.

We trained the Autoencoder as above, then holding all values the same except the latent dimension which we incrementally increased by steps of 10 features. We selected a size of 100 for the latent dimension since the larger the feature embedding the better the image reconstruction, measured by the minimum loss achieved. We required a feature

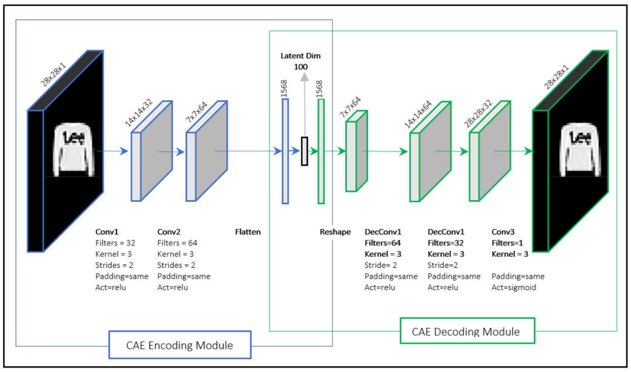


FIGURE 7. Autoencoder - a convolutional autoencoder containing the encoder to transform an image into a latent visual feature embedding with 100 features. Also the decoder which transforms a latent visual feature embedding back into an image.

TABLE 1. GMM hyper parameters for set of GMMs in baseline classifier.

Class	Reg CoVar	Comp. Num	Comp. Shape	BIC
0: T-Shirts	0.001	2	full	218599.8304
1: Trousers	0.00001	2	full	92074.9725
2: Pull Over	0.001	5	tied	215139.5721
3: Dress	0.01	5	tied	218177.7049
4: Coat	0.001	3	tied	209474.0943
5: Sandal	0.0001	2	full	271939.6885
6: Shirt	0.01	5	tied	225819.9335
7: Sneaker	0.001	2	tied	166274.3903
8: Bag	0.01	5	tied	293328.9243
9: Ankle Boot	0.01	3	tied	226256.1887

TABLE 2. Classifier Accuracy for balanced classes using 100% Training, Validation and Testing datasets. Acc: Accuracy, F1: Weighted F1 Score, CK: Cohen Kappa.

Data	Classifier	Acc %	F1	CK
Train	GMM	86.82	0.86	0.89
Valid	GMM	84.71	0.84	0.87
Test	GMM	85.57	0.85	0.87
Train	Softmax	97.97	0.98	0.98
Valid	Softmax	90.39	0.90	0.92
Test	Softmax	90.37	0.90	0.91

representation that can be decoded into a good image representation whilst still compressing the data to enable easier manipulation.

2) PROBABILISTIC MODEL

The probabilistic model is a Gaussian Mixture Model (GMM). A GMM was created for each symbolic class using encoded training images (100% dataset unless otherwise stated in an experiment). We evaluated all the combinations of the following hyper parameters; i) Number of mixture components: 1 to 5 inclusive, ii) Covariance type: Tied, Diagonal, Spherical & Full, and, iii) Non-negative regularization: 1.0e-2, 1.0e-3, 1.0e-4 and 1.0e-5. This resulted in 80 potential models per class. During GMM model creation, occasionally when regularisation was low the maximum likelihood estimation (MLE) for normal mixtures did not converge due to singularities or degeneracy. Any models which did not converge were automatically eliminated from our potential selection. The selected GMM had the minimum validation BIC score.

3) GMM-IL CLASSIFIER (*GMMs*)

Each learnt GMM was added to the set of GMMs to form the classifier *GMMs*. Table 1 shows the hyper parameters of the baseline set of GMMs. See Table 2 for this classifiers accuracy results.

4) BENCHMARK CLASSIFIER (*SOFTMAX*)

Our benchmark classifier (*Softmax*) is comprised of a deep learning network consisting of the same frozen encoder model, plus a dense layer with a Softmax activation. The hyper parameters were set to the same as described for the Autoencoder (see Section IV-E1).

V. RESULTS AND ANALYSIS

These experiments investigate the difference in performance between a multiple GMM head (*GMMs*) and the benchmark method of a single Softmax head (*Softmax*). Both classifiers use the same encoder with frozen weights trained on ten classes for all experiments except the experiment (Section V-D) where it was trained on six classes. The experiment (Section V-A) establishes a reference baseline. The next experiment evaluates the classifiers accuracy when using; small sample sizes during training (Section V-B) and when the sample size is imbalanced across classes (Section V-C). The experiment found in Section V-D reports the classifiers results when incrementally learning pairwise unseen classes.

A. CLASSIFIER BASELINE

The two classifiers were tested after building the models as described in Section IV-E3. The results for training, validation and testing are shown in Table 2. *Softmax* outperforms *GMMs* when 100% of each dataset is used and all classes are balanced.

B. SMALL SAMPLE SIZES

Focusing on small sample sizes a range of 5 to 20 (inclusive) samples are used. Stepping through each sample size all GMM models for both classifiers are retrained using the initial hyper parameter settings. As can be seen in Figure 8. *GMMs* perform with higher accuracy than *Softmax* for sample sizes smaller than 12.

C. IMBALANCED CLASSES

In the classification problem field, the scenario of imbalanced classes [36] appears when the numbers of samples that represent the different classes are very different. The minority classes are usually the most important concepts to be learnt, since they represent rare cases or because the data acquisition of these examples is costly. In this work three imbalanced ratio profiles are created and the classifiers weighted F1 Score are reported. The range 5 to 15 is selected, as in the 'Small

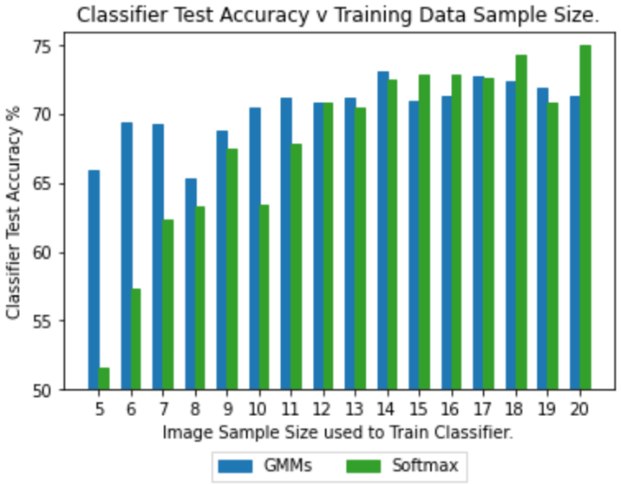


FIGURE 8. Classifier test accuracy for a training sample size of 5 to 20 inclusive.

Sample Sizes' experiment it was shown to be a range of interest. Using 5 as low and 15 as high 3 imbalanced class datasets were created. Imbalances that were covered are; (1) Extreme ratio difference of 1 class high and 9 classes low. (2) A 50:50 ratio difference of 5 class high, 5 class low and, (3) A stepped profile, Classes start at 5 samples and increment 1 sample until 14 samples. Both classifiers were retrained using the initial hyper parameters, GMM models where then fitted to new sample sizes based on the imbalanced experiment profile. Each Experiment was repeated 10 times with the class numbers rotating through the experiment profile. Figure 9 shows the mean accuracy and 95% confidence intervals per experiment and classifier type. Experiment 1,2 and 3 had p values of 0.000, 0.001 and 0.018 respectively. In all three experiments the (GMM) outperformed the (Softmax) when trained on sample sizes under 15.

D. CLASS INCREMENTAL LEARNING

Softmax classifiers learn all classes at once using all the training data. They do not perform as accurately when they are required to learn classes over time and have no access to previous training data.

We follow the benchmark method taken from Kolouri et al [37], and DeepMind, Google Research [38] who used the Split MNIST dataset to learn consecutive pairs. For our dataset this is pairs of clothes e.g., Pair 1: T-Shirt,Trousers, Pair 2: PullOver, Dress, Pair 3: Coat, Sandal, Pair 4: Shirt, Sneaker, Pair 5: Bag, Ankle Boot. We then make the following adjustments. We combine the first 3 pairs, which makes 6 classes trained in Task 1. The Autoencoder model is trained first using these 6 classes and the resulting encoder is frozen. Then, the 6 GMM Models are trained using their encoded images. This frozen encoder is then used for further tasks with just the GMM Models been trained. Pair 4 are used for Task 2 and Pair 5 used for Task 3. The reason Tasks contain 2 classes is to enable the *Softmax* to classify without having access to prior training data. For clarification the 3 Tasks were configured as follows:

1) Task 1 established the accuracy when the encoder was trained on 6 classes, the classifier heads (*GMM* and *Softmax*) were tested on 6 classes using 100% datasets. The classification was assigned to the class with the greatest probability/likelihood.
2) Task2 established the accuracy when the classifiers were trained as per Task1 with 2 further classes, the classifier heads (*GMM* and *Softmax*) were tested on 8 classes using 100% datasets. The classification was assigned to the class with the greatest probability/likelihood.
3) Task3 established the accuracy when the classifiers were trained as per Task2 with 2 further classes, the classifier heads (*GMM* and *Softmax*) were tested on all 10 classes using 100% datasets. The classification was assigned to the class with the greatest probability/likelihood.

Task1, Task2 and Task3 were repeated 10 times as the classes were rotated, the mean and 95% confidence values were calculated across all 10 combinations per classifier type. From the results shown in Figure 10 it can be seen that initially the *Softmax* is more accurate than the *GMMs*. However, after each Incremental Task, the *Softmax* accuracy decreases significantly more than the *GMM*. This shows the *GMMs* have a greater ability to retain class definitions than the *Softmax*.

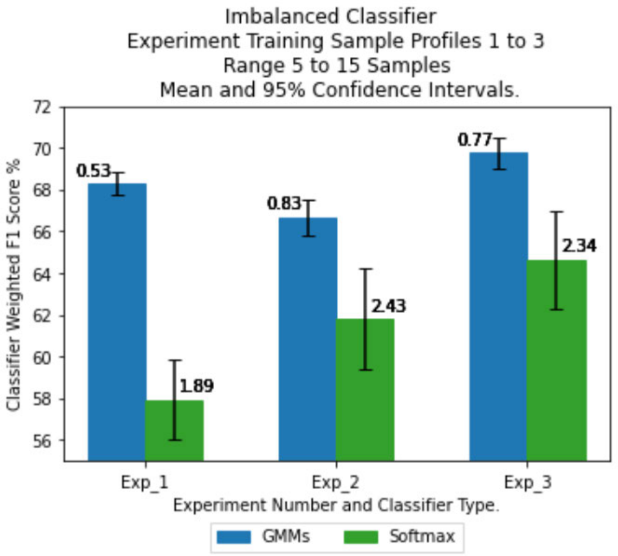


FIGURE 9. Classifier Weighted F1 Score for 3 Imbalanced Training Dataset Profiles. (Exp_1: 1 class n15 & 9 classes n5), (Exp_2: 5 classes n5 & 5 classes n15), (Exp_3: Classes start at 5 samples and increment to 14 samples.), Classes rotated 10 times. Mean and 95% confidence intervals shown.

VI. DISCUSSION

An architecture was created which enables transferred visual learning and the incremental addition of class definitions in

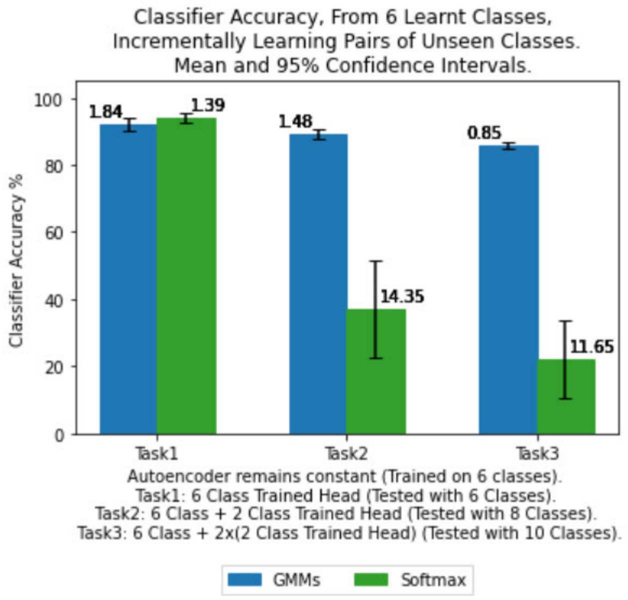


FIGURE 10. **Classifiers Incrementally Learning Three Tasks.**

the form of probabilistic models. The visual learning carried out used a smaller sample size than would ultimately be used to enable us to control the content of the visual features and verify that unseen classes could be learnt. This classifier's accuracy could be improved by making the following amendments.

- Autoencoder - Our main premise is that this model's accuracy is dependent on the quality of the latent space created by the autoencoder, by using a state of the art autoencoder the granularity and quality of the visual features will be improved and hence the discriminatory power of the classifier increased.
- Gaussian Mixture Models - Améndola et al [39] state that there is the possibility of more modes than means when Gaussians are combined. Further investigation needs to be carried out to optimise the accuracy of the GMM likelihood landscape for a set of GMMs.

VII. CONCLUSION AND FUTURE WORK

In conclusion, the proposed method creates a useful class representation where visual features are learnt using unsupervised training. Using these, independent probabilistic class definitions are trained which incorporate uncertainty. These representations are used within a classifier and benchmarked with an equivalent Softmax classifier which uses class relative probability. GMM-IL is found to be more accurate for sample sizes smaller than 12 images and more robust for three imbalanced class profiles in the same sample range. GMM-IL incrementally learns class definitions with no catastrophic forgetting issues which the Softmax benchmark exhibits. In conclusion, for a learning environment where only small sample sizes of new classes are available, this model shows good potential as a classifier. This paper describes the creation of symbolic definitions for items, this could be expanded to define descriptive adjectives, normal verbs and affordances through the training of GMMs on appropriate datasets. Once these symbolic definitions exist they could also be used to aid reasoning in Knowledge Graphs.

REFERENCES

[1] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, Jul. 2021.

[2] S. Zhang, G. Shen, J. Huang, and Z.-H. Deng, "Self-supervised learning aided class-incremental lifelong learning," 2020, *arXiv:2006.05882*.

[3] R. Kemker, M. McClure, A. Abitino, T. L. Hayes, and C. Kanan, "Measuring catastrophic forgetting in neural networks," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3390–3398.

[4] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6468–6477.

[5] A. Chaudhry, K. P. Dokania, T. Ajanthan, and H. S. P. Torr, *Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence* (Lecture Notes in Computer Science), vol. 11215. Cham, Switzerland: Springer, 2018.

[6] M. McCloskey and J. N. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychol. Learn. Motivat.*, vol. 24, pp. 109–165, Jan. 1989.

[7] K. James, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, and K. Milan, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[8] H. Liu, Y. Zhou, B. Liu, J. Zhao, R. Yao, and Z. Shao, "Incremental learning with neural networks for computer vision: A survey," *Artif. Intell. Rev.*, vol. 2022, pp. 1–33, Oct. 2022.

[9] Z. Li and D. Hoiem, "Learning without forgetting," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 614–629.

[10] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "ICaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5533–5542.

[11] A. Aich, "Elastic weight consolidation (EWC): Nuts and bolts," 2021, *arXiv:2105.04093*.

[12] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and A. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 11207, 2018, pp. 144–161.

[13] V. A. Terekhov, G. Montone, and J. K. O'Regan, "Knowledge transfer in deep block-modular neural networks," in *Proc. Conf. Biomimetic Biohybrid Syst.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 9222, Jul. 2015, pp. 268–279.

[14] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *Comput. Sci.*, vol. 1, pp. 1–14, Jun. 2016.

[15] J. Schwarz, J. Luketina, W. M. Czarnecki, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, "Progress & compress: A scalable framework for continual learning," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 10, 2018, pp. 7199–7208.

[16] J. Lee, D. Joo, H. G. Hong, and J. Kim, "Residual continual learning," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 4553–4560.

[17] G. I. Parisi, J. Tani, C. Weber, and S. Wermter, "Lifelong learning of spatiotemporal representations with dual-memory recurrent self-organization," *Frontiers Neurorobot.*, vol. 12, pp. 1–15, Nov. 2018.

[18] R. Kemker and C. Kanan, "Kemker Kanan—2018—Fearnet brain-inspired model for incremental lear.pdf," in *Proc. ICLR*, 2018, pp. 1–16.

[19] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, "Class-incremental learning: Survey and performance evaluation on image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Oct. 10, 2022, doi: 10.1109/TPAMI.2022.3213473.

[20] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Netw.*, vol. 113, pp. 54–71, May 2019.

[21] Y. Luo, L. Yin, W. Bai, and K. Mao, "An appraisal of incremental learning methods," *Entropy*, vol. 22, no. 11, pp. 1–27, 2020.

[22] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory," *Psychol. Rev.*, vol. 102, no. 3, pp. 419–457, Jul. 1995.

[23] D. Kumaran, D. Hassabis, and J. L. McClelland, "What learning systems do intelligent agents need? Complementary learning systems theory updated," *Trends Cognit. Sci.*, vol. 20, no. 7, pp. 512–534, Jul. 2016.

[24] A. B. U. Case, C. Chadebec, and L. J. Vincent, "Pythae: Unifying generative autoencoders in Python—A benchmarking use case," 2022, *arXiv:2206.08309*.

[25] G. H. V. D. Maaten, "Visualising data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 1, pp. 2579–2605, 2008.

[26] D. Reynolds, "Gaussian mixture models," in *Encyclopedia of Biometrics*. Boston, MA, USA: Springer, 2015, pp. 827–832.

[27] V. Maz'ya and G. Schmidt, "On approximate approximations using Gaussian kernels," *IMA J. Numer. Anal.*, vol. 16, no. 1, pp. 13–29, 1996.

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–22, Sep. 1977.

[29] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Design Implement.*, 2016, pp. 265–283.

[30] S. Haghighi, M. Jasemi, and S. Hessabi, "PyCM: Multiclass confusion matrix library in Python," *J. Open Source Softw.*, vol. 6, no. 4, pp. 2–3, 2018.

[31] R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.

[32] G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, vol. 1, no. 3, pp. 1–11.

[33] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2528–2535.

[34] P. D. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[35] I. Loshchilov and F. Hutter, "SGDR: S," in *Proc. ICLR*, 2017, pp. 1–16.

[36] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, Nov. 2016.

[37] S. Kolouri, N. Ketz, X. Zou, J. Krichmar, and P. Pilly, "Attention-based selective plasticity," 2019, *arXiv:1903.06070*.

[38] P. Kirichenko, M. Farajtabar, D. Rao, B. Lakshminarayanan, N. Levine, A. Li, H. Hu, A. G. Wilson, and R. Pascanu, "Task-agnostic continual learning with hybrid probabilistic models," in *Proc. ICML Workshop INNF*, 2021, pp. 1–23.

[39] C. Améndola, A. Engström, and C. Haase, "Maximum number of modes of Gaussian mixtures," *Inf. Inference, J. IMA*, vol. 9, no. 3, pp. 587–600, Sep. 2020.

PENNY JOHNSTON received the B.Eng. degree (Hons.) in engineering systems, computing and control and the M.Sc. degree in engineering management from Loughborough University, and the M.Sc. degree (Hons.) in big data from the University of Stirling, in 2018. She is currently pursuing the Ph.D. degree in AGI neuro-symbolic systems with the University of Stirling. She achieved C.Eng. Chartership with the Institution of Electrical Engineers. She has worked in industry for various large manufacturing companies, such as Siemens, Glaxo, AstraZeneca, and Kvaerner. Her research interests include visual perception, machine learning, and neuro-symbolic systems.

KEILLER NOGUEIRA received the B.Sc. degree in computer science from Universidade Federal de Viçosa, Brazil, in 2012, and the M.Sc. and Ph.D. degrees in computer science from Universidade Federal de Minas Gerais, Brazil, in 2015 and 2019, respectively. He is currently a Lecturer with the Division of Computing Science and Mathematics, University of Stirling, U.K. He has published several high-quality papers in leading journals and conferences. His research interests include deep and machine learning, pattern recognition, image processing, computer vision, and remote sensing.

KEVIN SWINGLER received the B.Sc. degree in computing and psychology from the University of Exeter and the M.Sc. and Ph.D. degrees from the University of Stirling. He is currently the Head of computing science and mathematics with the University of Stirling. His research interests include computer vision and machine learning, particularly applied in health settings.