Pattern Recognition 143 (2023) 109797

Contents lists available at ScienceDirect

Pattern Recognition

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

# One Shot Learning with class partitioning and cross validation voting (CP-CVV)

Jaime Duque-Domingo[a,*], Roberto Medina Aparicio[b], Luis Miguel González Rodrigo[b]

[a] *ITAP-DISA, University of Valladolid, Valladolid, Spain*
[b] *CARTIF Foundation, División de Sistemas Industriales y Digitales, Parque Tecnológico de Boecillo, Valladolid 47151, Spain*

ARTICLE INFO

*Article history:*
Received 20 January 2023
Revised 8 June 2023
Accepted 29 June 2023
Available online 3 July 2023

*Keywords:*
One Shot Learning
CP-CVV
Siamese
ConvNeXt
ViT
Regnet
Wide ResNet
ResNeXt
CVV
CNN
Ensemble models
Voting
Grocery product classification
CIFARFS

ABSTRACT

One Shot Learning includes all those techniques that make it possible to classify images using a single image per category. One of its possible applications is the identification of food products. For a grocery store, it is interesting to record a single image of each product and be able to recognise it again from other images, such as photos taken by customers. Within deep learning, Siamese neural networks are able to verify whether two images belong to the same category or not. In this paper, a new Siamese network training technique, called CP-CVV, is presented. It uses the combination of different models trained with different classes. The separation of validation classes has been done in such a way that each of the combined models is different in order to avoid overfitting with respect to the validation. Unlike normal training, the test images belong to classes that have not previously been used in training, allowing the model to work on new categories, of which only one image exists. Different backbones have been evaluated in the Siamese composition, but also the integration of multiple models with different backbones. The results show that the model improves on previous works and allows the classification problem to be solved, an additional step towards the use of Siamese networks. To the best of our knowledge, there is no existing work that has proposed integrating Siamese neural networks using a class-based validation set separation technique so as to be better at generalising for unknown classes. Additionally, we have applied Cross-Validation-Voting with ConvNeXt to improve the existing classification results of a well-known Grocery Store Dataset.

## 1. Introduction

One of the biggest challenges in the field of computer vision is the classification of images of which there is only one per category. For example, in supermarkets, it is not easy to create a dataset that includes hundreds of images per product. Moreover, every time a new product is added, it would be necessary to retrain the model with the entire dataset. One Shot Learning (OSL) techniques aim to solve this type of problem, producing models that are able to classify images with only one example per category. Among the techniques used to implement OSL, Siamese neural networks have been shown to be very effective.

The normal training of a Siamese network is used to learn whether two images belong to the same category or not through comparison. In contrast to the traditional training of this type of models, the training of Siamese neural networks can be carried out using different classes for training, validation and testing. This technique allows the model to be trained using hundreds of images of known classes and that knowledge is used for comparison with images of unknown classes. Similarly, humans are better able to recognise people with known facial features because our brains have been trained primarily with such patterns [1].

Training a Siamese neural network to detect whether two images belong to unknown categories is ambitious, because such a model has not been trained with data from such classes. One of the problems with these systems is the lack of generalisation. It is in this aspect where the models can benefit from the Cross Validation Voting (CVV) technique [2], previously applied to improving the classification results of normal Convolutional Neural Networks (CNN). In this paper, we show how the state-of-the-art convolutional networks embedded in Siamese networks can be improved using a new class partitioning method, hereafter CP-CVV (Class Partitioning based Cross Validation Voting). This method applies class partitioning instead of data partitioning, selecting *k* val-

[*] Corresponding author.
*E-mail address:* jaime.duque@uva.es (J. Duque-Domingo).

https://doi.org/10.1016/j.patcog.2023.109797

idation groups with different classes in order to better generalise the model.

The training of a Siamese network using different classes is interrupted by early-stopping when the results of the validation set start to deteriorate. The model is then evaluated against a test set, where the classes are different from those used during training and validation. A problem with this training method is that the model obtained by early-stopping could overfit against the validation set (known as validation overfitting). The consequence of this is that the model does not respond well to new data, representing a generalisation problem. In the CP-CVV method, as there are *k* validation sets formed by non-intersecting categories, different models are trained to generalise to different situations. By pooling the models using voting techniques, an improvement in the generalisation is achieved.

Although model integration techniques have been used with some neural networks, to the best of our knowledge, there is no existing work that has proposed an integration of Siamese neural networks and, more specifically, a class-based validation set separation technique to make the model better able to generalise to unknown classes. Integration methods usually use different models, and if they are of the same type, they use validation sets with classes similar to those used in the training and other training/validation sets. Our approach is completely novel, since we train a model of the same type by performing the separation of the validation sets by classes. Moreover, we have performed the tests using state-of-the-art networks, such as ConvNeXt [3].

In addition to the experiments carried out to show the improvement in Siamese networks, the behaviour of ConvNeXt [3] has been evaluated using CVV [2], showing how it improves the classification results to date on the Grocery Store Dataset [4].

In our study, we focus on the OSL problem rather than the Few Shot Learning problem (FSL), as it is a particularly interesting case because we often have only one image per category and it is not possible to obtain more. In FSL, there is more than one image per category.

The present paper is structured as follows: Section 2 explores the state-of-the-art of the technologies considered in this paper. Section 3 describes the procedure that has been carried out. In Section 4, the different experiments and results obtained with the proposed method are reported. An overall discussion of the results obtained is set out. Finally, Section 5 notes the advantages and limitations of the system presented and suggests future developments.

## 2. Overview of related work

One Shot Learning represents a learning paradigm where only one item per category is available during the classification process. For example, in the grocery sector, it is interesting to classify products by means of a vision system from a single image per product. Furthermore, some OSL systems do not require the retraining of the model each time an image is added. For example, face recognition systems [5] usually work by using vectors obtained from the images that are compared to search for the closest face. These systems are not retrained with new people every time someone new is added to the database, as this would incur a high computational cost for each new person added.

Different strategies have been developed to solve the OSL problem. Until the development of deep learning, many techniques used probabilistic approaches. For example, in [6], the probability that an object belongs to a class was calculated by analysing image features that had been useful in classifying objects of the same type. Descriptor extractors based on characteristic points, such as SIFT, SURF or ORB, also relied on matching the obtained points with the points of known objects. These methods, however, do not generalise well when there are significant differences in the objects. The authors of [7] used a hybrid approach for supermarket product classification that combined feature-based matching and deep learning.

The generation of synthetic data has begun to be considered for retraining classification models. The traditional data augmentation allows the number of images to be increased by simple transformations (translations, rotations, illumination changes, deformations, etc.). This technique is integrated with many data generators that feed images into the training methods. More recently, the *Generative Adversarial Networks* (GANs) [8] are able to generate synthetic images using deep networks. They need to learn how to generate new unknown images from a generator trained with known images and a discriminator forces the new images to be different from previously known images. In [9], the authors proposed a GAN architecture to augment the training set of grocery products. They performed kNN recognition on a database consisting of a single reference image per product. However, one of the problems with this technique is that it usually requires retraining the model on the basis of new images that are added to the dataset.

Siamese neural networks [10] compare the output features of two networks, usually convolutional, to infer whether two images belong to the same category or not. The comparison is carried out using the feature vectors obtained before the last classification layers. Each of these sub-networks, called backbones, shares the model and weights. Although such networks were first used by [11] in a signature verification work, it is only in recent years that they have shown their potential. They have been used for a variety of problems in vision, such as object tracking [12], chromosome classification [13], diagnosis of COVID-19 patients [14], object segmentation [15], face recognition [16] and even face spoofing detection [17].

FSL can be treated by algorithms that follow two different approaches. On the one hand, in the inductive setting, training data are available but not test data. Inductive methods seek to generate a function or a model that returns the category of a test image that has never been seen before. In the OSL problem, once the training of the model is done, an example of each category would be available. Recent examples of this approach are Prototypical networks (ProtoNets)[18], Attentional Constellation Nets [19] PEMnE-NCM [20] or HCTransformers [21]. Our method also corresponds to an OSL inductive method in which we are completely unaware of the test set during training. In the transductive setting, less restrictive than the inductive case, training and unlabelled test data are available. The methods can obtain extra information about the test data distribution to make better predictions. Many current methods are transductive because sometimes it is easy to obtain test samples even if labelling is complicated. Some transductive FSL methods are PT+MAP+SF+SOT [22], PEMnE-BMS [20], the Illumination Augmentation + PT+MAP [23], SIB [24], P-M-F [25], BAVARDAGE [26] or EASY 3xResNet12 [27]. Some of these methods can also work in inductive setting.

Regarding the inductive methods, ProtoNets [18] learn a metric space in which classification can be carried out by calculating distances to prototype representations of each class. Compared to recent approaches for low-data learning, they reflect a simpler inductive bias that is beneficial in this data-limited regime, and they achieve excellent results. ProtoNets can work with different problems, from Zero Shot Learning, through OSL (one sample per class), to FSL (several samples per class). Attentional Constellation Nets [19] perform cell feature clustering and encoding with a dense part representation. Then, they use an attention mechanism to model the relationships between the cell features. They combine different constellation branches with convolutional feature maps to increase the awareness of object parts. PEMnE-BMS [20] uses a feature extractor trained using a generic dataset. Then, the features are preprocessed using PEME (Power, Euclidian normalization, Mean subtraction, Euclidean normalization) to better align with a Gaussian distribution. Finally, they are directly fed to a Nearest Class Mean Classifier (PEMnE-NCM). The authors also presented a transductive setting where the features are processed through an optimal transport inspired algorithm using self-distillation and Boosted Min-Size Sinkhorn (BMS). In HCTransformers [21], the authors proposed hierarchically cascaded transformers that exploit intrinsic image structures through spectral tokens pooling to reduce the ambiguity between foreground content and background noise. In addition, the learnable parameters are optimized through latent attribute surrogates to benefit from the rich visual information in image-label pairs.

Regarding the transductive methods, the authors of BAVARDAGE [26] proposed a new clustering method based on Bayesian Variational inference, further improved by Adaptive Dimension Reduction based on Probabilistic Linear Discriminant Analysis. They sought to take better account of uncertainty in estimation due to missing data, as well as better statistical properties of the clusters associated with each class. The authors of PT+MAP+SF+SOT [22] defined a module called Self-Optimal-Transport (SOT), which allows the transformation of features in a nonparametric and differentiable way and can capture high-level relationships between data points. It can transform global feature information to make it more differentiable in case-specific problems such as clustering, few-point learning, and person re-identification. In [23], the authors presented the Illumination Augmentation method. It uses a neural network architecture called Separating-Illumination Network (Sill-Net) that learns to separate illumination features from images. Then, the augmentation module takes the illumination features to augment the support samples. They aligned their method with the pipeline of PT+MAP in a transductive way for the FSL problem. SIB [24] uses the empirical Bayes formulation for multi-task learning, leveraging the unlabelled query set in addition to the support set to generate a more powerful model for each task. The authors of EASY 3xResNet12 [27] created a model that works with an ensemble of convolutional backbones to extract the features. These are concatenated, processed and evaluated in two settings: Nearest class mean classifier (NCM) if in inductive setting or a soft K-means algorithm in transductive. Finally, P-M-F [25] pretrains a Vision Transformer with the unlabelled external data using self-supervised loss. Then, it trains the model using simulated labels with a ProtoNet loss.

In relation to grocery products, the authors of [28] recently applied a Siamese network to capture the relationships between iconic and natural images in the Grocery Store Dataset [4]. They evaluated several Siamese models with different CNNs, obtaining the best results with a DenseNet-169 backbone [29]. One of the problems with this approach is that it uses iconic images, which may not be faithful to a real photograph of a product.

Grocery product recognition has many applications, such as monitoring food habits [30]. In recent years, several datasets have been created for grocery stores, such as the MVTec D2S dataset [31], the Retail Product Checkout dataset (RPC) [32], or the Freiburg groceries dataset [33]. These datasets focus on the problem of object detection rather than classification. The Grocery Store Dataset [4] contains images of grocery products, classified into fine and coarse categories. It contains 5125 images of 81 different types of fruit, vegetables and carton items (e.g., milk, juice or yoghurt). In addition, there are 43 coarse classes, grouping some categories. The authors separated the test set in order to properly compare different models and architectures. A classification baseline was also provided, where the authors evaluated several models, obtaining a test accuracy of 85.0% using a DenseNet [34] with SVM. These results were first surpassed by [35], where a stacking model of two ResNeXt-101 obtained 90.80% test accuracy; then by the authors of [36], who obtained 93.48% test accuracy using an ensemble of different networks (ResNet-101, ResNet-152, DenseNet-121, DenseNet-169 and DenseNet-201); and finally by [2], who obtained 94.41% test accuracy using a soft-voting CVV model based on 5 classifiers ResNeXt-101, WideResNet-101 and EfficientNet-B7.

In relation to the backbones used to compose our Siamese nets, we have evaluated several recent models, such as ResNeXt-101 [37], Wide Residual Networks (WRNs) [38], EfficientNet-B7 [39], RegNet X_32gf [40], ViT-L-32 [41] and ConvNeXt Large [3].

ResNeXt [37] is an architecture that replaces the 3x3 convolutions within the ResNet model with clustered 3x3 convolutions. The ResNeXt bottleneck block splits a single convolution into multiple smaller parallel convolutions. ResNeXt uses aggregation instead of concatenation in the original Inception-ResNet block. Wide Residual Networks (WRNs) [38] consider the problem that each fraction of a percent of improved accuracy costs almost double the number of layers. The authors proposed a novel architecture in which they decreased the depth and increased the width of the residual networks. This architecture deals with the problem of diminishing feature reuse, which makes the training of residual networks slow. EfficientNet [39] seeks a balance between the number of parameters and accuracy. This multi-objective neural architecture optimises both accuracy and FLOPS, similar to MNASNet [42]. EfficientNet-B7 scales depth, width and resolution from EfficientNet-B0 using a composite coefficient. In RegNet [40], a *design space design* principle was presented. They conducted population based experiments on hundreds of models, looking at how parameters and settings affect different criteria. They introduced RegNet as an effective design space according to those principles.

Vision Transformers (ViT) [41] are based on transformers originally designed for NLP tasks. While the CNNs use pixel convolutions, the ViT divides images into visual fixed-size patches, correctly embeds each patch, and includes positional embedding as input to the transformer's encoder. This transformer uses a self-attention layer, able to enhance some parts of the input data while diminishing other parts, focusing on the most important areas of the image. ViT usually requires a large dataset, so transfer learning is usually used as a starting point.

Although the transformer-based models managed to outperform ImageNet [43] classification results with respect to CNNs in recent years, ConvNeXt [3] has been able to bring convolutional models to the top again, using certain features inherited from the ViT models. It uses depthwise convolutions, which are similar to the weighted sum operation in self-attention, and *Gaussian Error Linear Unit* (GELU) activation functions, similar to ViT. In addition, it uses larger kernel sizes and an inverted bottleneck design that reduces the parameters, thus increasing the performance.

## 3. Analysis of the system

We propose a new technique called CP-CVV, based on a modification of the CVV technique [2], previously used to improve the classification of CNNs, to enhance the training of Siamese neural networks. When CVV is applied to CNNs, the training data are divided into $k$ different validation slots, with the remaining data not used for validation in each slot being chosen as the training data for that slot. A single classifier type, such as ResNeXt-101 [37], is trained $k$ times with each different validation set, and the outputs of the models are finally combined using soft and hard-voting techniques.

In CP-CVV, the validation sets for each $k$ slot are selected by distributing the $n$ classes into $k$ validation slots (see Fig. 1). That is, for each of the $k$ trainings of a model, the validation set will consist of $\approx n/k$ classes. The order of the classes is randomised before the validation slots are allocated. In this way, we prevent potentially related classes from entering into a single training.

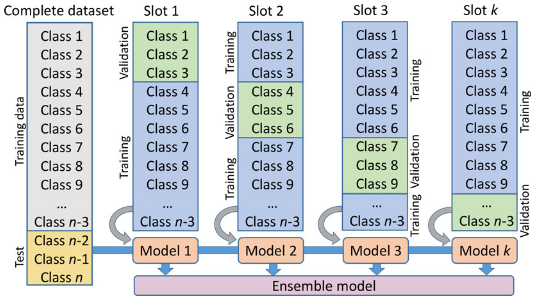


**Fig. 1.** Distribution of classes in $k$ slots.

Our CP-CVV model integrates $k$ Siamese neural networks using a soft/hard voting mechanism. In contrast to how other Siamese nets are trained, in our approach, each of the $k$ independent networks is trained independently with different training and validation classes. Fig. 2 shows the scheme of the model during inference. The model receives two images, corresponding to a positive pair if the images correspond to the same class and negative otherwise. The pair of images is fed into each of the $k$ Siamese nets, formed by a backbone of the same type, which produces a vector of features. The weights of this backbone are similar within the same Siamese, but different among the $k$ models. The output feature vector of each backbone is multiplied by an element-wise multiplication. Three dense layers are then added, integrated with dropout and batch normalisation. The first of the dense layers uses a Rectified Linear Unit (ReLU) activation function, while the output of the second dense layer uses sigmoid activation. Finally, a last dense layer is responsible for the classification using another sigmoid activation function. Unlike other Siamese networks, where the connection of the backbone features is done by calculating the Euclidean distance, several dense hidden layers are used in this case. As can be seen in Fig. 2, the output of each Siamese network approximates the value to 0 or 1, depending on whether the pair has been classified as positive or negative. In the integration of multiple classifiers by hard-voting, we accumulate the number of positive and negative pairs detected, with the final output producing the largest number of postings. In the case of soft-voting, the output value is accumulated among the different classifiers and then divided by the total number of classifiers. If the result is greater than $\frac{1}{2}$, then it will be a positive pair, and negative otherwise.

Let $\tau$ be the set that includes all the classes of the dataset, $\lambda$ the set of classes used for training and $\beta$ the set of classes for testing. Let $T_i$ and $V_i$ be the training and validation sets corresponding to the slot $i$ and $k$ the number of slots used in CP-CVV. These sets must verify Eqs. 1 to 4.

$$\tau = \lambda \cup \beta \tag{1}$$

$$\lambda = \bigcup_{i=1}^{k} V_i \tag{2}$$

$$\bigcap_{i=1}^{k} V_i = \emptyset \tag{3}$$

$$\left[T_i = \lambda - V_i\right] \forall i \in k \tag{4}$$

The classes of a training slot $i$ are those used by all the other slots in the validation, as shown in Eq. 5.

$$\left[T_i = \bigcup_{j=1}^{k} V_{j:j\neq i}\right] \forall i \in k \tag{5}$$

Similarly, a validation slot, $i$, is composed of the intersection of the classes of the other training slots, as shown in Eq. 6.

$$\left[V_i = \bigcap_{j=1}^{k} T_{j:j\neq i}\right] \forall i \in k \tag{6}$$

Each $k$ model is trained with their respective training and validation slot. This data distribution means that the ensemble model can be generalised better for different situations, avoiding the validation overfitting. The inference is carried out using voting. For an input sample, $x$, $p_i(x)$ is the sigmoid output value given by the Siamese network $i$. In our experiments, we have seen that a Siamese net with one sigmoid output gave better results than a network with two outputs; one to show that the two images belonged to the same category and one for the opposite case. In the sigmoid output case, the output takes the value 0 when the images belong to the same class and 1 otherwise. In Eq. (7), soft voting is obtained by accumulating the output values given by all the classifiers. $w_i$ is a weight associated with each classifier $i$, $\frac{1}{k}$ in our case. If the result is greater than $\frac{1}{2}$, then the images belong to different categories and the global output is set to 1.

$$S(x) = \begin{cases} 1 & \text{if } \left[\sum_{i=1}^{k} w_i \cdot p_i(x)\right] > \frac{1}{2} \\ 0 & otherwise \end{cases} \tag{7}$$

Hard voting requires prior binarisation of the probability, as shown in Eq. (8). The output of each classifier approaches 0 or 1, depending on whether the images belong to the same category or not. Then, as shown in (9), the output is obtained by accumulating the binary values of each class $j$. As in soft voting, $w_i$ is $\frac{1}{k}$ in our case, and the output approaches 0 or 1, depending on the sum.

$$b_i(x) = \begin{cases} 1 & \text{if } p_i(x) > \frac{1}{2} \\ 0 & otherwise \end{cases} \tag{8}$$

$$H(x) = \begin{cases} 1 & \text{if } \left[\sum_{i=1}^{k} w_i \cdot b_i(x)\right] > \frac{1}{2} \\ 0 & otherwise \end{cases} \tag{9}$$

### 3.1. Classification problem

Siamese nets allow us to infer whether two images belong to the same class. However, in a real classification application, what would be of interest would be to correctly classify images into a particular category. So, in a supermarket, you may want to add new classes and have the system take care of cataloguing images of incoming products among those classes. For each class, there would only be one catalogue image.

In CP-CVV, the outputs of several Siamese nets are mathematically comparable as they represent whether or not two images belong to the same category. In the classification problem, we analyse the cumulative probability that an image belongs to each of the possible categories. In other words, we draw $k \cdot c$ inferences from the Siamese nets, obtaining a matrix with $k$ rows and $c$ columns. A cell of the matrix represents the probability that an image belongs to class $c$ in the $k$ slot. If we accumulate the column values of that cell and divide by $k$, we obtain the probability that an image belongs to that class according to all the slots.

Let $P_c$ be the cumulative result of adding the sigmoid outputs of the different Siamese nets for a test class $c$, where $c \in \beta$. Therefore, $P_c$ is the value of a test image belonging to a category. This is evaluated by selecting one random image per category. Let $p_{ci}$ be the sigmoid output of the classifier $i$ with an image from the category $c$. For soft voting, $P_c$ is obtained by (10)

$$P_c(x) = \sum_{i=1}^{k} w_i \cdot p_{ci}(x) \tag{10}$$

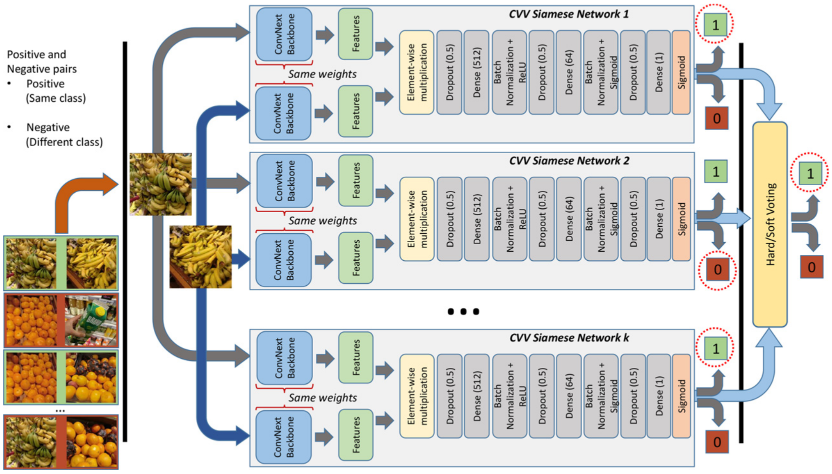


**Fig. 2.** Diagram of the proposed solution.

In order to find the most similar category, we keep the one that is closest to 0. This is achieved with the $arg_c min$ function, as shown in Eq. (11). This function returns the winning class.

$$C_{soft}(x) = arg_c min\left[P_c(x)\right] \tag{11}$$

In the case of the hard voting classification, $p_{ci}$ is previously binarised, as shown in (12); where only the output with the lowest class value is set to 0 for an estimator.

$$b_{ci}(x) = \begin{cases} 0 & \text{if } c = arg_c min[p_{ci}(x)] \\ 1 & otherwise \end{cases} \tag{12}$$

Then, $P'_c$ is obtained by adding the binarised value in this case, see Eq. (13), and the winning category is given by (14).

$$P'_c(x) = \sum_{i=1}^{k} w_i \cdot b_{ci}(x) \tag{13}$$

$$C_{hard}(x) = arg_c min\left[P'_c(x)\right] \tag{14}$$

We show below the experiments carried out to demonstrate that the proposed method improves the result obtained by Siamese networks trained without CP-CVV.

## 4. Experiments and results discussion

We have used the Grocery Store Dataset [4], a dataset of supermarket products that allows us to evaluate the performance of the proposed method, as it has many classes and can be separated into test classes and different validation slots.

The experiment consisted of evaluating how Siamese networks improve using the new CP-CVV method. For this, we performed a cross-validation against the test data, choosing $s = 4$ different test sets and applying the CP-CVV to the rest of the data. In this way, we verified that the model responds correctly to different test sets. Table 1 shows the list of test classes for each slot $s$. It is important to note that the evaluation of the 4 test sets is different from the validation slots used during training. We have performed four training runs with CP-CVV and four evaluations and have shown average values.

CP-CVV has been applied with $k = 5$ by distributing the training data into 5 validation slots, as previously explained. In Fig. 3, we can see the distribution of the classes in $s = 0$ with $k = 5$ validation slots. To train the $k = 1$ Siamese network, the classes of its slot are used as validation, while the rest of the classes are used as training data. It is important to point out that one aspect is the cross validation applied to testing, whereby the aim is to see that the new CP-CVV method works well for different selected $s$ test slots; and another different aspect is the application of CP-CVV itself, which is carried out on $k$ validation slots.

To evaluate the effectiveness of the CP-CVV method applied to Siamese neural networks, several state-of-the-art backbones have been used to implement the Siamese nets: ResNeXt-101 [37], Wide Residual Networks (WRNs) [38], EfficientNet-B7 [39], RegNet X_32gf [40], ViT-L-32 [41] and ConvNeXt Large [3].

Each of the models was trained using a CP-CVV with $k = 5$ (independent validation sets of the classes). Fig. 4 shows the training during 100 epochs of the $s = 0$ and $k = 0$ model for these estimators, where we can appreciate that the models that converge the best and the fastest are EfficientNet-B7 and ConvNeXt Large. These plots have been shown to visualise a similar training of 100 epochs. However, as early-stopping is applied, the training is completed in a smaller number of times. The graph up to 100 epochs has been generated by training the model during those epochs in order to show graphs with similar ranges. When using early-stopping, we evaluate the validation drop for 10 epochs. Thus, for example, ConvNeXt and ResNeXt converge in less than 40 epochs.

The training started from a transfer learning of the weights of the backbones previously trained against ImageNet [43]. The images were then normalised with respect to the mean values obtained from ImageNet. In addition, data augmentation has been used, performing transformations that include rotation (20°), translation (20%), scaling (20%) and shearing (20%). Random flips (50%) have also been applied. An Adam optimiser, with a learning rate of 0.0004, and the binary crosentropy loss were used during the training.

As can be seen, the evaluation of a CP-CVV Siamese model based on a particular backbone requires quite a few training sessions, specifically $s \cdot k$. However, this is only for the most realistic evaluation of the model improvement. Since the test classes are completely independent of the models, for a real problem, it would be sufficient to carry out $k$ trainings.

The training of the models was carried out on a deep learning server with two Xeon Gold 6230R processors, two 48GB Nvidia RTX A6000 GPUs, and 768GB of RAM. The average training time for each of the models was 720 minutes. However, as several training sessions were parallelised due to GPU capacity, the training of the $k$ models of each type and each $s$ slot were reduced. In total, for $s = 4$ and $k = 5$, we trained 120 models ($s \cdot k \cdot 6$ different models), taking approximately 21,600 minutes (15 days) to be completed.

Table 2 shows that the CP-CVV model applied to the training of multiple Siamese networks with a similar backbone improves

**Table 1**
Test class distribution for cross validation with $s = 4$ slots.

| $s = 0$ | $s = 1$ |
|---|---|
| Arla-Ecological-Sour-Cream | Alpro-Blueberry-Soyghurt |
| Brown-Cap-Mushroom | Arla-Mild-Vanilla-Yoghurt |
| Carrots | Cabbage |
| God-Morgon-Orange-Juice | God-Morgon-Apple-Juice |
| God-Morgon-Red-Grapefruit-Juice | God-Morgon-Orange-Red-Grapefruit-Juice |
| Granny-Smith | Honeydew-Melon |
| Lime | Nectarine |
| Mango | Oatly-Natural-Oatghurt |
| Oatly-Oat-Milk | Red-Bell-Pepper |
| Passion-Fruit | Red-Delicious |
| Sweet-Potato | Satsumas |
| Tropicana-Golden-Grapefruit | Tropicana-Juice-Smooth |
| Yellow-Onion | Zucchini |
| $s = 2$ | $s = 3$ |
| Alpro-Fresh-Soy-Milk | Alpro-Blueberry-Soyghurt |
| Alpro-Vanilla-Soyghurt | Arla-Lactose-Medium-Fat-Milk |
| Banana | Arla-Natural-Mild-Low-Fat-Yoghurt |
| Beef-Tomato | Cantaloupe |
| God-Morgon-Orange-Red-Grapefruit-Juice | Carrots |
| Kiwi | Conference |
| Orange-Bell-Pepper | Garlic |
| Papaya | Honeydew-Melon |
| Solid-Potato | Pomegranate |
| Tropicana-Mandarin-Morning | Tropicana-Juice-Smooth |
| Watermelon | Vine-Tomato |
| Yellow-Bell-Pepper | Yoggi-Vanilla-Yoghurt |
| Yoggi-Strawberry-Yoghurt | Zucchini |

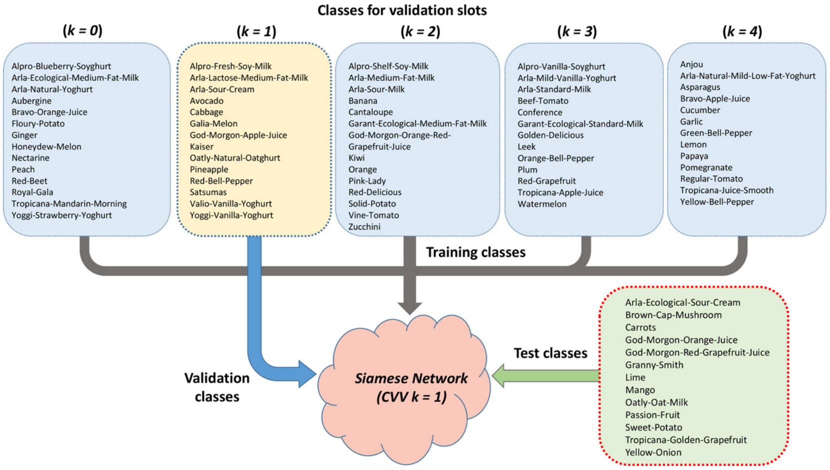
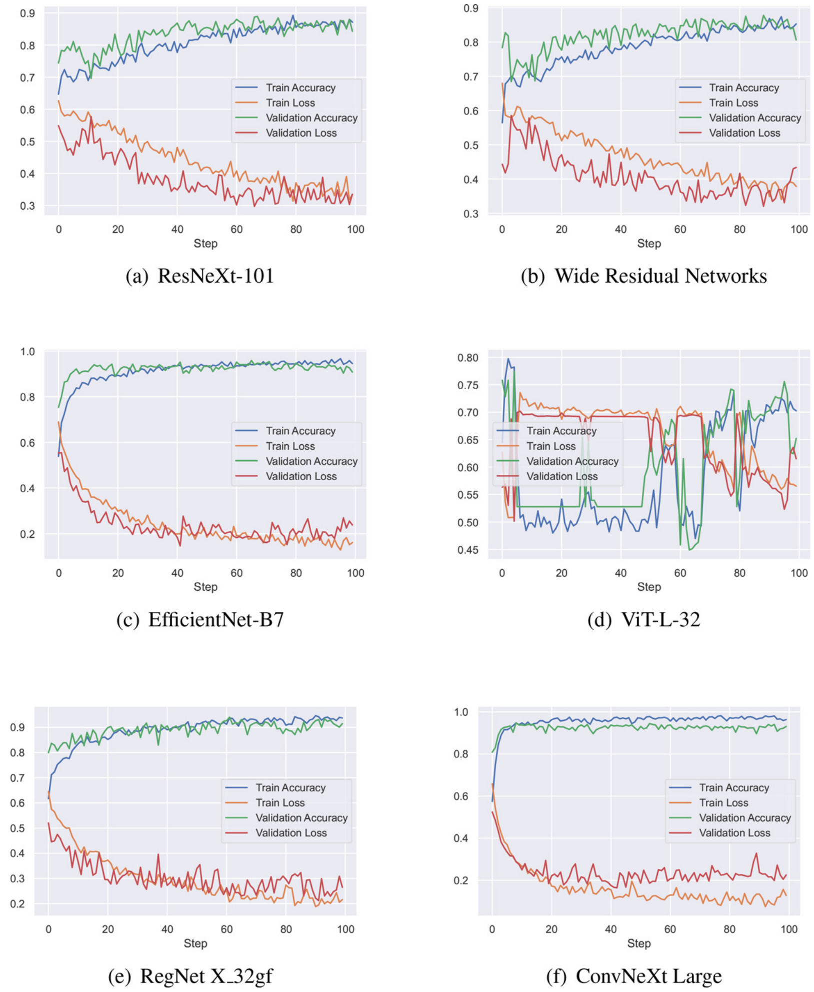


**Fig. 3.** Separation and training for CP-CVV ($k = 1$).

**Table 2**
CP-CVV improvement with different classification Siamese networks for $k = 5$ (% except for loss). HV: Hard Voting. SV: Soft Voting.

| Backbone\Metric | Test accuracy without CP-CVV | Test accuracy (CP-CVV HV) | Test accuracy (CP-CVV SV) | Loss (without CP-CVV) | Loss (CP-CVV HV) | Loss (CP-CVV SV) |
|---|---|---|---|---|---|---|
| ResNeXt-101 | 0.8490 | 0.8893 | 0.8966 | 3.3570 | 2.0504 | 2.9732 |
| Wide-ResNet-101 | 0.8510 | 0.8993 | 0.9051 | 3.2538 | 1.9845 | 2.9188 |
| ViT-L-32 | 0.7770 | 0.8216 | 0.8248 | 4.7435 | 3.0337 | 4.2682 |
| EfficientNet-B7 | 0.8825 | 0.9380 | 0.9392 | 2.1585 | 1.4383 | 1.7875 |
| RegNet_x_32gf | 0.8718 | 0.9111 | 0.9184 | 2.6174 | 1.7371 | 2.4807 |
| ConvNeXt_large | 0.8961 | 0.9498 | 0.9486 | 1.9314 | 1.3044 | 1.6533 |
| Global CP-CVV (All models with k = 5) | | 0.9390 | 0.9469 | | 1.9247 | 2.6803 |
| Selective CP-CVV (ConvNeXt_large + EfficientNet-B7) | | 0.9483 | 0.9526 | | 1.3714 | 1.7204 |
| Selective CP-CVV (ConvNeXt_large + RegNet_x_32gf + EfficientNet-B7) | | 0.9548 | 0.9560 | | 1.4933 | 1.9739 |



(a) ResNeXt-101

(b) Wide Residual Networks

(c) EfficientNet-B7

(d) ViT-L-32

(e) RegNet X_32gf

(f) ConvNeXt Large

**Fig. 4.** Training of Siamese networks with different backbone for CP-CVV $k = 0$ (100 epochs).

remarkably. We can observe how the method, in both its soft-voting (SV) and hard-voting (HV) versions, improves the results of all Siamese using a single model. As an example, a Siamese net with a ConvNeXt_large backbone (0.8961) is improved with 5 estimators in both hard (0.9498) and soft-voting (0.9486). The results are the average of the cross validation of $s$ slots.

We have also evaluated models that integrate different backbones, including one that combines all models (ResNeXt-101, Wide-ResNet-101, ViT-L-32, EfficientNet-B7, RegNet_x_32gf and ConvNeXt_large), one that integrates the top two models (EfficientNet-B7 and ConvNeXt_large) and one that integrates the top three models (EfficientNet-B7, RegNet_x_32gf and ConvNeXt_large). The integration uses $k = 5$ estimators per backbone.

The results show that the best results are obtained with the selective model that integrates the three best models (0.9560 in CP-CVV SV). The integration of these three models accumulates the probability of the 15 associated models in the case of soft-voting. Note that each model has been trained 5 times with the different validation sets. For the case of hard-voting, instead of accumulating probabilities, 1 or 0 is accumulated, depending on whether the result is greater than $\frac{1}{2}$, as explained above. Transformers and attention models, such as ViT-L-32, do not provide good results in our experiments. These models usually require very large datasets and do not always respond well to the generalisation problem, as we found here. The best result using a single model integrated with the new CP-CVV is provided by ConvNeXt (0.9498). As we can see from the results, it is important to highlight the fact that the new CP-CVV technique oriented to Siamese networks always improves the result with respect to the traditional Siamese.

Although, in most cases, it performs better with soft-voting than with hard-voting, the error seems to be superior in soft-voting. This is due to the way the output is calculated, since in SV it is an accumulation of probabilities, which usually leads to a higher error.

Table 3 shows the comparison of the results with the previous work that explored the combination of a neural network-based backbone and a LOMO descriptor within the same Siamese net. That work used the same dataset and the same evaluation technique. We can see how the new CP-CVV method improves the results of that work using only the ResNeXt-101 backbone. If we consider other backbones, such as ConvNext_large or the selective CP-CVV ensemble (ConvNeXt_large + RegNet_x_32gf + EfficientNet-B7), the results are greatly outperformed.

### 4.1. Classification problem

To carry out the experiments related to the classification problem, a random image is selected from all the test categories, and a random image from each possible category is chosen for cataloguing. The selected image is compared with each of the selected images for each category. The result is the most similar category. However, at any given moment, the model may predict that two or more categories are similar, but only the one with the lowest value is kept, since the positive pairs are those close to 0. In ad-

**Table 3**
CP-CVV improvement over comparable work (%).

| Technique | Test accuracy |
|---|---|
| Siamese with ResNeXt-101 | 0.8490 |
| Siamese with ResNeXt-101 + LOMO backbone [35] | 0.8820 |
| CP-CVV SV Siamese with ResNeXt-101 backbone | 0.8966 |
| CP-CVV HV Siamese with ConvNeXt_large backbone | 0.9498 |
| Selective CP-CVV SV Siamese (ConvNeXt_large + RegNet_x_32gf + EfficientNet-B7) | 0.9560 |

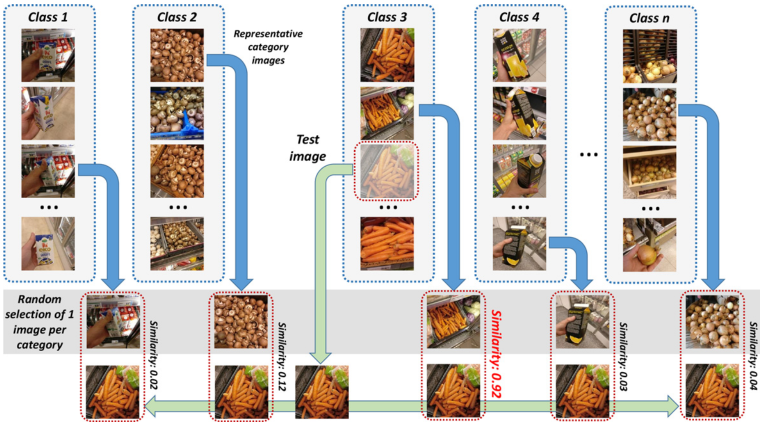


**Fig. 5.** Selection of images to solve the classification problem.

dition, the TOP-2 value is computed, which shows whether any of the possible categories are correct, i.e., if the model has catalogued an image in two possible categories, and the real category is one of them. Sometimes, there are certain categories that represent images of objects with a certain similarity. The image categorisation loop is repeated for the entire set of images to be evaluated.

Fig. 5 shows how the image selection process is carried out for the evaluation. For each of the images in the test set, one image is randomly selected from each of the test categories. Then, the pairs formed by the image being evaluated and each of the random images are introduced and evaluated in the Siamese net, or the set of Siamese networks, using the new CP-CVV. The category with the lowest value of similarity is the winning class, with which the evaluated image is associated. In the new CP-CVV, a Siamese $k$ net is applied to each pair and the output values are accumulated in the case of soft-voting, choosing the category with the lowest cumulative value, as explained above. In the case of hard-voting, the output of each Siamese is previously binarised, checking the output with the lowest class value given for an estimator, as previously explained. Then, the values are accumulated among the $k$ siameses, as is done in soft-voting, and the winner is the category with the lowest cumulative value.

Table 4 shows the classification result for the test images, i.e., the classes that have not been used to train the Siamese networks. This inference technique is the most realistic one used to solve classification problems using Siamese networks, and the results obtained show its potential for application in a real problem. In this case, it can be seen that the model which uses the combination of the ConvNeXt_large and EfficientNet B7 classifiers by means of the new CP-CVV obtains the best classification results. In 79.26% of the cases, the category is correct. We have to take into account the fact that there can be a lot of differences between images of the same category. For example, a milk carton can be upside down in one image and pictured frontally in another. This makes the classification a challenge and other methods, such as those looking for image key points, do not work well in these cases. The experiments have been repeated 5 times and the values shown are the average.

In addition to the evaluation of CP-CVV with Siamese nets, two experiments with Prototypical networks (ProtoNets) [18] and Attentional Constellation Nets [19] have been carried out. Although ProtoNets can deal with the problem of zero-shot learning, our experiments focus on one-shot-learning. This implies that the number of labelled examples per class in the support set is equal to one ($n_s = 1$). We have also set up a query size of 25 images per class ($n_q = 25$). On the other hand, the network has been trained to classify 13 classes (number of classes in a classification task: $n_{way} = 13$). We have also used data augmentation and the same test set to compare the results. Although the results obtained with ProtoNet are significantly worse than most of our models, it is important to mention that it has converged quickly. In 18 epochs and 134 minutes, it achieved an 89.03% training accuracy (peak value). Regarding the Attentional Constellation Nets, we have also used the same test and parameters: $n_s = 1$, $n_q = 25$ and $n_{way} = 13$. This model has completely converged in 110 minutes, even less than ProtoNet, and 56 epochs (train acc: 99.96%, val acc:71.33%). Our model, parallelised, requires about 720 minutes of training. The test accuracy of the Attentional Constellation Net was 76.44%, only surpassed by some of our more complex ensemble models.

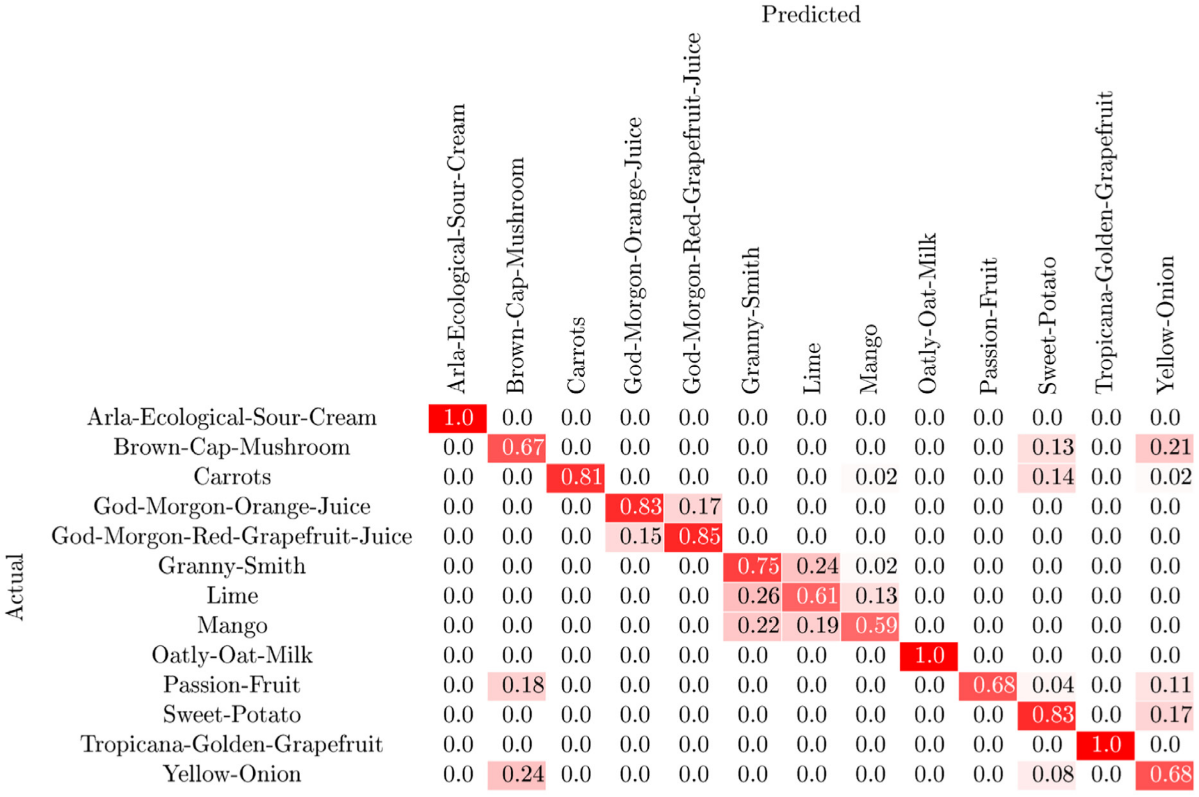
Table 5 shows the confusion matrix of the selective model with ConvNeXt_large and EfficientNet-B7, using soft-voting and $k = 5$ estimators. There are categories where the images have some similarities and the confusion matrix shows a decrease in accuracy, such as *Mango* and *Lime*, or *Yellow-Onion* and *Brown-Cap-Mushroom*. This confusion matrix has been developed taking into account the classes of slot $s = 0$ (see Table 1). The system would obviously work better if the images were always taken from a similar position, but it shows how our system gives good results even in these situations of high variability in images of the same category.

We have also evaluated the method with a general dataset, CIFAR-FS [44] (CIFAR-100 few-shots), which is randomly sampled from CIFAR-100. This dataset is divided into training, validation and test classes. We have combined the training and validation classes and then divided them into $K = 4$ slots according to CP-CVV. We then trained 4 models with a ConvNeXt_small backbone

**Table 4**
Results of using the CP-CVV method with Siamese neural networks for the classification problem in OSL for $k = 5$ (% except for loss). HV: Hard Voting. SV: Soft Voting (Average from 5 repetitions of the test).

| Backbone / Metric | Test accuracy (without CP-CVV) | Test accuracy (CP-CVV HV) | Test accuracy (CP-CVV SV) | TOP-2 accuracy (without CP-CVV) | TOP-2 accuracy (CP-CVV HV) | TOP-2 accuracy (CP-CVV SV) |
|---|---|---|---|---|---|---|
| ResNeXt-101 | 0.5313 | 0.6120 | 0.6484 | 0.7765 | 0.7857 | 0.8525 |
| Wide-ResNet-101 | 0.5369 | 0.6332 | 0.6535 | 0.7742 | 0.7857 | 0.8364 |
| ViT-L-32 | 0.4415 | 0.4525 | 0.4995 | 0.6590 | 0.5853 | 0.7258 |
| EfficientNet-B7 | 0.6834 | 0.7111 | 0.7419 | 0.8894 | 0.8871 | 0.9171 |
| RegNet_x_32gf | 0.5313 | 0.6152 | 0.6378 | 0.8134 | 0.8065 | 0.8548 |
| ConvNeXt_large | 0.7355 | 0.7825 | 0.7899 | 0.8963 | 0.8963 | 0.9194 |
| Global CP-CVV (All models with k = 5) | | 0.7406 | 0.7544 | | 0.8963 | 0.9055 |
| Selective CP-CVV (ConvNeXt_large + EfficientNet-B7) | | **0.7926** | 0.7899 | | 0.9355 | 0.9378 |
| Selective CP-CVV (ConvNeXt_large + RegNet_x_32gf + EfficientNet-B7) | | 0.7677 | 0.7576 | | 0.9240 | 0.9332 |
| ProtoNet [18] | 0.5964 | | | 0.7700 | | |
| Attentional Constellation Net [19] | 0.7644 | | | 0.8224 | | |

**Table 5**
Confusion matrix of the selective CP-CVV model (ConvNeXt_large + EfficientNet-B7).



Predicted

| Actual | Arla-Ecological-Sour-Cream | Brown-Cap-Mushroom | Carrots | God-Morgon-Orange-Juice | God-Morgon-Red-Grapefruit-Juice | Granny-Smith | Lime | Mango | Oatly-Oat-Milk | Passion-Fruit | Sweet-Potato | Tropicana-Golden-Grapefruit | Yellow-Onion |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Arla-Ecological-Sour-Cream | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Brown-Cap-Mushroom | 0.0 | 0.67 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.13 | 0.0 | 0.21 |
| Carrots | 0.0 | 0.0 | 0.81 | 0.0 | 0.0 | 0.0 | 0.0 | 0.02 | 0.0 | 0.0 | 0.14 | 0.0 | 0.02 |
| God-Morgon-Orange-Juice | 0.0 | 0.0 | 0.0 | 0.83 | 0.17 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| God-Morgon-Red-Grapefruit-Juice | 0.0 | 0.0 | 0.0 | 0.15 | 0.85 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Granny-Smith | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.75 | 0.24 | 0.02 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Lime | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.26 | 0.61 | 0.13 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Mango | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.22 | 0.19 | 0.59 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Oatly-Oat-Milk | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Passion-Fruit | 0.0 | 0.18 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.68 | 0.04 | 0.0 | 0.11 |
| Sweet-Potato | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.83 | 0.0 | 0.17 |
| Tropicana-Golden-Grapefruit | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 |
| Yellow-Onion | 0.0 | 0.24 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.08 | 0.0 | 0.68 |

for 10 epochs. Being smaller images, 32x32 pixels, the training has been relatively fast, about 120 minutes sequentially. We have also used data augmentation. The Siamese nets achieved approximate accuracy values of 0.82 for validation and 0.97 for training. Next, we evaluated the model against test sets of 5 classes ($n_{way} = 5$) and on the OSL problem ($n_{shots} = 1$). We performed 5 runs, classifying 1,000 random images at a time, and calculated the mean values. Table 6 shows the classification result with the CP-CVV SV and CP-CVV HV methods and their comparison with other models. It should be noted that CP-CVV is open to combine different models, so it is possible that these values could be improved by adding some other combined classifier. As some models can work in both inductive and transductive modes, the table reflects under which setting the experiment with the best results was conducted.

Our method works in inductive setting, which means that it is a more restrictive method than transductive methods. In the performance comparison of FSL with CIFAR_FS, some methods are inductive and some are transductive. Under the same experimental conditions, it should be noted that transductive methods have some advantage in using unlabelled images of the test cases themselves because they can obtain extra information about the test data distribution to make better predictions. Within the inductive methods, our method is easy to implement and obtains promising results with CIFAR_FS, achieving higher values than the other methods presented.

Finally, a different kind of experiment was carried out to evaluate the CVV method with the most modern convolution network, ConvNeXt [3]. The result of the experiment has improved the latest

**Table 6**
Comparison of test accuracy (%) with other current models against the CIFAR_FS 100 model ($n_{way} = 5, n_{shots} = 1$).

| Model | Setting | Test accuracy |
|---|---|---|
| PT+MAP+SF+SOT [22] | Transductive | 0.8994 |
| PEMnE-BMS [20] | Transductive | 0.8844 |
| Illumination Augmentation + PT+MAP [23] | Transductive | 0.8773 |
| BAVARDAGE [26] | Transductive | 0.8735 |
| EASY 3xResNet12 [27] | Transductive | 0.8716 |
| CP-CVV SV with ConvNeXt_small backbone (ours) | Inductive | **0.8550** |
| P-M-F with ViT [25] | Transductive | 0.8430 |
| CP-CVV HV with ConvNeXt_small backbone (ours) | Inductive | **0.8386** |
| SIB [24] | Transductive | 0.8000 |
| HCTransformers [21] | Inductive | 0.7889 |
| EASY 3xResNet12 [27] | Inductive | 0.7620 |
| PEMnE-NCM [20] | Inductive | 0.7484 |

**Table 7**
Comparison of different classification models.

| Model / Metric | Test accuracy | Balanced test accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| DenseNet-169 with SVM [4] (baseline) | 0.8500 | | | | |
| Model with ResNeXt-101 [35] (140 epochs) | 0.9080 | 0.9209 | 0.9250 | 0.9210 | 0.9230 |
| Cascade Model with ResNeXt-101 [35] | 0.9200 | 0.9306 | 0.9350 | 0.9310 | 0.9330 |
| Ensemble "C" of different classifiers [36] | 0.9348 | | 0.9498 | 0.9452 | 0.9446 |
| Soft CVV with ensemble of 5 ResNeXt-101, 5 EfficientNet B7 and 5 Wide ResNet-101 [2] | 0.9441 | 0.9555 | 0.9580 | 0.9560 | 0.9570 |
| Soft CVV with 5 ConvNeXt Large (our experiment) | 0.9580 | 0.9696 | 0.9720 | 0.9700 | 0.9710 |

classification results obtained against the Grocery Store Dataset [4]. As the test set of this dataset was selected by the authors themselves, the results are comparable under the same conditions as other existing works. Table 7 shows the comparison of this model trained with CVV and other models working on the same dataset. The authors of the Grocery Store Dataset established a baseline with a DenseNet-169 that was combined with an SVM classifier using the feature vector. In [35], a stacking model of two ResNeXt-101 was evaluated, obtaining 92% balanced test accuracy. In [36], the authors evaluated different ensembles of networks, achieving 94.98% balanced test accuracy. Their best result was obtained using a hard-voting approach, integrating the following models: ResNet-50, ResNet-101, ResNet-152, EfficientNet-B1, DenseNet-121, DenseNet-169 and DenseNet-201. Finally, the best results to date were obtained using the CVV technique on a model composed of 5 ResNeXt-101, 5 EfficientNet B7 and 5 Wide ResNet-101 [2]. In our work, we evaluated this method on the ConvNeXt Large network, achieving the best results to date (95.80% test accuracy / 96.96% balanced test accuracy).

The experiments carried out have shown how the new CP-CVV method improves the behaviour of Siamese networks, independently of the backbone used. Moreover, due to the nature of the training itself, the resulting model allows promising results to be obtained in the classification of images into categories where we only have one example. Furthermore, the CVV method has also been used with the latest convolutional network to improve classification results on a well-known grocery product dataset.

There is a limitation to the CP-CVV method that depends on the number of slots created. The method starts to offer improved results from $k = 2$ and goes up to a maximum value of $k$, $k = 5$ in this article. From that moment on, if we increase $k$, the results will be inverted. The method follows a parabola of the accuracy as a function of $k$ and it is necessary to try with different $k$ until the optimal value is obtained. This search is costly, since for each specified $k$ we have to multiply the training time by $k$ ($k \cdot t$). In addition, the inference time will also be multiplied by $k$. If we use more resources (more GPU memory, multiple GPUs, distributed processing, etc.), these times will be reduced. On the other hand, for values with $k < 4$, it is recommended to limit the data distribution in validation. If we do not limit the slot size, an imbalance between the amount of data in training and in validation will occur.

Additionally, solving the classification problem itself has an associated cost. In our method, $c$ forward steps must be carried out for an image, where $c$ is the number of classes (this is because they are Siamese networks). In addition, if we have $k$ models, the time will be that of the inference of $k \cdot c$ models. Although it may seem quite a lot, the results of the method are promising and again, depending on resources, the inferences can be parallelised. Naturally, the higher the number of evaluation classes, the worse the results and the longer the inference time.

Finally, it is worth mentioning that, although we could select validation slots with some intersection, and this would certainly also improve the results of the individual model, in CP-CVV we propose that as we are dealing with a Siamese network problem that must separate images of the same or different categories, we must train models with different classes in validation so as to be able to respond to different cases that have similar nature; thus giving us the ability to decide whether or not two images belong to the same category.

## 5. Conclusions

We have presented a system that integrates different Siamese neural networks using a modification of the CVV method, called CP-CVV, based on class-oriented CVV partitioning. This technique trains multiple classifiers, based on the same backbone, with different validation sets whose intersection is the empty set. However, as the problem in One Shot Learning is to be able to classify unknown classes, the validation partitioning is performed using different classes from those used during training, but which in turn do not overlap between the $k$ validation sets. The models are integrated using soft and hard voting techniques. Finally, the models are evaluated using another test set with classes that are different from the training and validation ones.

The results of the experiments show that the combined model using CP-CVV is able to improve the previous results obtained with the Grocery Store Dataset. In addition, the method has been evaluated with respect to the classification problem itself. Siamese networks allow us to tell whether two images belong to the same class or not, but an additional step is necessary to classify an image among a set of possible categories. The classification results also offer promising results, considering the large difference between images belonging to a similar category, which would make it unfeasible to use other methods based on obtaining key points. As an additional experiment, all the published classification results of this dataset have been improved using the latest CVV-integrated convolution network, ConvNeXt.

Different current backbones for Siamese networks have been evaluated (ResNeXt-101, Wide ResNet-101, ViT-L-32, EfficientNet-B7, RegNet_X-32 and ConvNext_large), and how the new CP-CVV technique always improves the performance of individual Siamese networks has been demonstrated. The best combined classifier uses the 5 ConvNext_large and 5 EfficientNet-B7 classifiers. We have also evaluated our method against a Prototypical network and an Attentional Constellation Net, which have required less training time, but have produced lower classification results. Our model has also been evaluated with CIFAR_FS, showing that even as an inductive method it competes with some of the best transductive methods.

The main advantage of the CV-CPP model is that it is relatively simple to use and improves the results with all the cases we have evaluated. It allows the results of any type of Siamese network, used mainly for the OSL problem, to be boosted. The main limitation of CV-CPP is that it requires parallel inference, so it usually requires one or more GPUs with a larger memory. It also involves longer training time, as it is necessary to train several models. Depending on the trade-off between accuracy and inference time, the CV-CPP model is configurable and we can select a different number of sub-models to integrate.

Our further research will aim to improve the unknown class classification problem itself. Although 79% is a promising figure, it is far from the values sought by companies and industries. We aim to solve this problem by using additional techniques that could also consider the case of FSL, in which it would be necessary to have a few images of each category in order to improve the results.

## Credit Author Statement

J.D.D. contributed to the entire work, designing the methodology, experiments, software, analysing the results and preparing the paper. R.M.A. contributed to the work with the funding acquisition, designing the experiments and analysing the results. L.M.G.R. contributed with the funding acquisitions, monitoring the work progress.

## Declaration of Competing Interest

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

## Data availability

We have used a public dataset. We do not need to share it because it is public.

## Acknowledgements

This publication has been partially funded by the project “5R-Cervera Network in robotic technologies for smart manufacturing”, contract number CER-20211007, under “Centros Tecnológicos de Excelencia Cervera” programme funded by “The Centre for the Development of Industrial Technology” (CDTI) and by the “Instituto para la Competitividad Empresarial de Castilla y León” (Project CCTT3/20/VA/0003) through the line “2020 Proyectos I+D orientados a la excelencia y mejora competitiva de los CCTT”, co-financed with FEDER funds.

## References

[1] E.A. Phelps, Faces and races in the brain, Nat. Neurosci. 4 (8) (2001) 775–776.

[2] J.D. Domingo, R.M. Aparicio, L.M.G. Rodrigo, Cross validation voting for improving CNN classification in grocery products, IEEE Access 10 (2022) 20913–20925.

[3] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A convnet for the 2020s, arXiv preprint arXiv:2201.03545 (2022).

[4] M. Klasson, C. Zhang, H. Kjellström, A hierarchical grocery store image dataset with visual and semantic labels, in: IEEE Winter Conference on Applications of Computer Vision (WACV), 2019.

[5] M. Wang, W. Deng, Deep face recognition: a survey, Neurocomputing 429 (2021) 215–244.

[6] L. Fe-Fei, et al., A Bayesian approach to unsupervised one-shot learning of object categories, in: Proceedings Ninth IEEE International Conference on Computer Vision, IEEE, 2003, pp. 1134–1141.

[7] W. Geng, F. Han, J. Lin, L. Zhu, J. Bai, S. Wang, L. He, Q. Xiao, Z. Lai, Fine-grained grocery product recognition by one-shot learning, in: Proceedings of the 26th ACM international conference on Multimedia, 2018, pp. 1706–1714.

[8] A. Mikołlajczyk, M. Grochowski, Data augmentation for improving deep learning in image classification problem, in: 2018 international interdisciplinary Ph.D. workshop (IIPhDW), IEEE, 2018, pp. 117–122.

[9] A. Tonioni, L. Di Stefano, Domain invariant hierarchical embedding for grocery products recognition, Comput. Vision Image Understanding 182 (2019) 81–92.

[10] G. Koch, R. Zemel, R. Salakhutdinov, et al., Siamese neural networks for one-shot image recognition, ICML Deep Learning Workshop, volume 2, Lille, 2015.

[11] J. Bromley, J.W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, R. Shah, Signature verification using a siamese time delay neural network, Int. J. Pattern Recognit Artif Intell. 7 (04) (1993) 669–688.

[12] X. Sun, G. Han, L. Guo, H. Yang, X. Wu, Q. Li, Two-stage aware attentional siamese network for visual tracking, Pattern Recognit 124 (2022) 108502.

[13] S. Jindal, G. Gupta, M. Yadav, M. Sharma, L. Vig, Siamese networks for chromosome classification, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 72–81.

[14] M. Shorfuzzaman, M.S. Hossain, MetaCOVID: a siamese neural network framework with contrastive loss for n-shot diagnosis of COVID-19 patients, Pattern Recognit 113 (2021) 107700.

[15] X. Gong, X. Liu, Y. Li, H. Li, A novel co-attention computation block for deep learning based image co-segmentation, Image Vis Comput 101 (2020) 103973.

[16] A. Holkar, R. Walambe, K. Kotecha, Few-shot learning for face recognition in the presence of image discrepancies for limited multi-class datasets, Image Vis Comput 120 (2022) 104420.

[17] M. Pei, B. Yan, H. Hao, M. Zhao, Person-specific face spoofing detection based on a siamese network, Pattern Recognit 135 (2023) 109148.

[18] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, Adv Neural Inf Process Syst 30 (2017).

[19] W. Xu, Y. Xu, H. Wang, Z. Tu, Attentional constellation nets for few-shot learning, in: International Conference on Learning Representations, 2021.

[20] Y. Hu, S. Pateux, V. Gripon, Squeezing backbone feature distributions to the max for efficient few-shot learning, Algorithms 15 (5) (2022) 147.

[21] Y. He, W. Liang, D. Zhao, H.-Y. Zhou, W. Ge, Y. Yu, W. Zhang, Attribute surrogates learning and spectral tokens pooling in transformers for few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9119–9129.

[22] D. Shalam, S. Korman, The self-optimal-transport feature transform, arXiv e-prints (2022) arXiv–2204.

[23] H. Zhang, Z. Cao, Z. Yan, C. Zhang, Sill-Net: feature augmentation with separated illumination representation, arXiv preprint arXiv:2102.03539 (2021).

[24] S.X. Hu, P.G. Moreno, Y. Xiao, X. Shen, G. Obozinski, N.D. Lawrence, A. Damianou, Empirical Bayes transductive meta-learning with synthetic gradients, arXiv preprint arXiv:2004.12696 (2020).

[25] S.X. Hu, D. Li, J. Stühmer, M. Kim, T.M. Hospedales, Pushing the limits of simple pipelines for few-shot learning: external data and fine-tuning make a difference, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9068–9077.

[26] Y. Hu, S. Pateux, V. Gripon, Adaptive dimension reduction and variational inference for transductive few-shot classification, arXiv preprint arXiv:2209.08527 (2022).

[27] Y. Bendou, Y. Hu, R. Lafargue, G. Lioi, B. Pasdeloup, S. Pateux, V. Gripon, Easy: ensemble augmented-shot y-shaped learning: state-of-the-art few-shot classification with simple ingredients, arXiv preprint arXiv:2201.09699 (2022).

[28] G. Ciocca, P. Napoletano, S.G. Locatelli, Iconic-based retrieval of grocery images via siamese neural network, in: Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part II, Springer International Publishing, 2021, pp. 269–281.

[29] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.

[30] B. Sainz-De-Abajo, J.M. García-Alonso, J.J. Berrocal-Olmeda, S. Laso-Mangas, I. De La Torre-Díez, Foodscan: food monitoring app by scanning the groceries receipts, IEEE Access 8 (2020) 227915–227924.
[31] P. Follmann, T. Bottger, P. Hartinger, R. Konig, M. Ulrich, MVTec D2S: densely segmented supermarket dataset, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 569–585.
[32] X.-S. Wei, Q. Cui, L. Yang, P. Wang, L. Liu, Rpc: a large-scale retail product checkout dataset, arXiv preprint arXiv:1901.07249 (2019).
[33] P. Jund, N. Abdo, A. Eitel, W. Burgard, The freiburg groceries dataset, arXiv preprint arXiv:1611.05799 (2016).
[34] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, K. Keutzer, DenseNet: implementing efficient convnet descriptor pyramids, arXiv preprint arXiv:1404.1869 (2014).
[35] J. Duque Domingo, R. Medina Aparicio, L.M. González Rodrigo, Improvement of one-shot-learning by integrating a convolutional neural network and an image descriptor into a siamese neural network, Applied Sciences 11 (17) (2021) 7839.
[36] M. Leo, P. Carcagnì, C. Distante, A systematic investigation on end-to-end deep recognition of grocery products in the wild, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 7234–7241.
[37] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1492–1500.
[38] S. Zagoruyko, N. Komodakis, Wide residual networks, arXiv preprint arXiv:1605.07146 (2016).
[39] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.
[40] I. Radosavovic, R.P. Kosaraju, R. Girshick, K. He, P. Dollár, Designing network design spaces, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10428–10436.
[41] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).
[42] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, Q.V. Le, Mnas-Net: Platform-aware neural architecture search for mobile, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2820–2828.
[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.
[44] L. Bertinetto, J.F. Henriques, P.H.S. Torr, A. Vedaldi, Meta-learning with differentiable closed-form solvers, arXiv preprint arXiv:1805.08136 (2018).

**Jaime Duque Domingo** holds a Ph.D. in Systems and Control Engineering from the UNED (2018), a Master's Degree in Software Engineering and Computer Systems from the UNED (2014) and a Master of Education (Mathematics) from the University Isabel I (2018). He obtained a Bachelor's Degree in Computer Engineering from the University of Valladolid in 2011. He is currently Assistant Professor at the University of Valladolid. For 18 years, he worked on the development of complex IT projects for the private sector, both in Spain and abroad. During the last few years, he has focused on the academic world, being professor at private (UEMC) and public universities and participating in different projects. He has published a book on Computer Vision with Machine Learning, 13 articles in SCI-JCR indexed journals, in the first and second quartile, and 14 papers in national and international conferences. His field of activity is mainly focused on computer vision and robotics, specializing in deep neural networks. He has been a visiting professor at Carnegie Mellon University in Pittsburgh and has obtained five awards in his research career: INFAIMON 2015, IARIA-ICWMC 2017, INFAIMON 2018, Extraordinary PhD Award 2020 and Innovadores 2022.

**Roberto Medina Aparicio** is an Industrial Engineer and Doctor from the University of Valladolid. He has worked as a researcher at CARTIF since 2005, where he has combined research work and industrial development. He began his professional career in the Robotics and Machine Vision Division and is currently part of the Industrial and Digital Systems Division. He has extensive experience in research projects involving machine vision, sensorization, 3D reconstruction, and distributed computing. He has published 8 scientific articles in scientific journals of impact and in various conferences, as well as has participated in 1 international patent. He has worked on many research projects, both national and European. Currently working on research projects related to computer vision based on deep neural networks, such as I-visart ("New artificial vision methodologies for the visual inspection of highly reflective and textured surfaces") and Agrovis ("Artificial Vision for products/processes in the agrifood sector"). He is also the project manager at CARTIF of CERVERA network in robotic technologies for smart manufacturing (5R).

**Luis Miguel González Rodrigo** received the Automatic Control and Systems Engineer's Degree in 2003 and the M.Eng. in the Automatic Control and Systems program in 2007, both from the University of Valladolid (Spain). He has been Project Manager and R&D engineer in the Robotics and Computer Vision Division at CARTIF Technological Centre for over 17 years. He has extensive experience in the design, development and management of industrial projects associated with quality control and process automation through the use of artificial vision techniques, mainly in the automotive sector. He has worked on many research projects, both national and European, such as Trex ("Extended Range Robot Enabling Technologies for the Flexible Factory") and Vapex ("Novel Vapor Analyzer for the Detection of Explosives in Airports Passengers Checkpoints"), where CARTIF will define and develop an automatic robotic manipulator to manage and transport the analyzer filters. Currently working on research projects related to computer vision based on deep neural networks, such as I-visart ("New artificial vision methodologies for the visual inspection of highly reflective and textured surfaces") and Agrovis ("Artificial Vision for products/processes in the agrifood sector").