This software implements the experiments relative to One-Shot Learning mentioned in the paper “One Shot Learning with Class Partitioning and Cross Validation Voting (CP-CVV)”, published in Pattern Recognition (2023).

Authors:

Jaime Duque Domingo

ITAP-DISA

Departamento de Ingeniería de Sistemas y Automática

University of Valladolid, Valladolid, Spain

eMail: [jaime.duque@uva.es](mailto:jaime.duque@uva.es)

Roberto Medina Aparicio, Luis Miguel González Rodrigo

CARTIF Foundation, División de Sistemas Industriales y Digitales, Parque Tecnológico de Boecillo, 47151 Valladolid, Spain

Abstract (from our Pattern Recognition paper)

One Shot Learning includes all those techniques that allow to classify images using a single image per category. One of its possible applications is the identification of food products. In the case of a food store, it is interesting to record a single image of each product and to be able to recognize 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.

This code shows the classification experiments carried out in our paper on Pattern Recognition for the One Shot Learning problem. It uses a new Siamese network training technique, called CP-CVV, which uses the combination of different models trained with different classes. The separation of the validation classes has been done in such a way that each of the combined models is different to avoid overfitting with respect to validation. Unlike normal training, the test images belong to classes that have not been previously used in training, which allows the model to work with 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, a further step towards the use of Siamese networks.

Requirements:

Torchvision >= 0.12 (for the ConvNeXt backbone). with CUDA support.

Pytorch > 1.11.0 with CUDA support.

Tensorboard

nvidia-ml-py3

Steps:

1. Download the Grocery Store Dataset used by the experiments from:

<https://github.com/marcusklasson/GroceryStoreDataset.git>

1. Modify **step\_0\_moveFiles.py** (variable **PATH\_DATASET**) to point to the dataset directory.
2. Run **python** **step\_0\_moveFiles.py.** *In this process, we unify all data under the whole directory and generate s = 5 folders for cross-validation. Within each folder there will be another k = 5 folders to implement CP-CVV. (check the Pattern Recognition paper)*
3. Modify step\_1\_training\_int.sh to point to the dataset directory (variable **DATASET\_ORIGEN**).
4. Run **step\_1\_training.sh**. *This process runs 5 times* ***step\_1\_training\_int.sh****, one for each s slot of cross-validation.* ***step\_1\_training.sh*** *launches the k=5 trainings for each CP-CVV slot for all the neural backbones trained. Before running each training, it verifies the amount of GPU available. For each training, it runs* ***step\_1\_training.py****, the process that implements the training.*

*This process will take several days of training depending on available resources. A total of s=5 x k=5 x 6 backbones are launched. That is, 120 different trainings for the Siamese composed of different backbones: ResNeXt-101, Wide Residual Networks, EfficientNet-B7, ViT-L-32, RegNet X 32gf, ConvNeXt Large. Please check the Pattern Recognition paper.*

1. Modify **step\_2\_evaluate.py** (variable **PATH\_DATASET**) to point to the dataset directory.
2. Run **python** **step\_2\_evaluate.py**. *This process evaluates the performance improvement of Siamese nets from an individual point of view and combined with CP-CVV. The evaluation of the classification with One-Shot-Learning is not carried at this point. In the paper, s=4 cross-validation slots were used. Although 5 are trained in the past, 5 are used. It prints different tables and data used in the paper.*
3. Modify **step\_3\_classification\_evaluation.py** (variable **PATH\_DATASET**) to point to the dataset directory.
4. Run **python** **step\_3\_classification\_evaluation.py**. *This process carries out the classification evaluation for One-Shot-Learning. It transforms the Siamese network into a regular classification problem. An evaluation of each image is performed with the rest of the pairs to see if it correctly classifies the image.*