Measuring the effectiveness Few-shot learning for traffic sign recognition

Alessandro Bevilacqua
Department of Computer Science
University of Bari Aldo Moro
Bari (BA), Italy
a.bevilacqua11@studenti.uniba.it

Mattia Colucci
Department of Computer Science
University of Bari Aldo Moro
Bari (BA), Italy
m.colucci76@studenti.uniba.it

I. Abstract

Nowadays, self-driving vehicles is one of the main application of Artificial Intelligence (AI) which has been most widely used in daily-life scenarios. These vehicles are based on many AI models among which there are traffic sign recognition systems. Different proposals have been provided for these systems, most of which rely on the usage of large datasets such as GTSRB (dataset of German traffic signs). Collecting a big amount of data is one of the main problems of Machine Learning models, which need lots of evidences in order to being able to generalize well. The cost of collecting more data than available can be one of the main aspects to consider during the development of these systems. Furthermore, using a big dataset exposes the model to possible Adversarial attacks, such as poisoning attacks, especially in case of out-source training and testing. For these reasons, in this work we investigate the application of a state-of-art (S.O.T.A.) Few-shot Learning (FSL) model, called LDC, on the traffic sign recognition domain. The aim of this work is to compare few-shot performance with non-few-shot ones by using different kinds of S.O.T.A. and baseline models on multiple datasets. The comparison is performed statistically using a Paired T-test.

II. Introduction

Few-shot learning has become one of the main frontiers of current researchers in Artificial Intelligence and Machine Learning. Especially Computer Vision has been highly impacted by these kinds of methods. Deep Learning models are able to excel in different domains in which a large dataset is available, Despite this, they struggle in cases where a small amount of samples is provided. This is due to the fact that these models need lots of evidences in order to correctly train their parameters and achieve good performance. In these scenarios, most of the models sin in generalizing and poorly perform on unseen data [6].

Having a lack of samples is a common problem in Machine Learning, especially in many domains such as medicine and security [6]. Finding and/or crafting more samples in order to have a bigger dataset, is not a trivial task, and may lead in having huge costs that need to be considered in a project.

Among the breadth of domains in which AI is used the most, autonomous vehicles is one of the most important. Currently, in some countries of the world, autonomous vehicles are used for transportation without the need of the driver. Just think of Waymoo, one of the pillar agency in this domain, which has many active cars for autonomous transportation in cities like San Francisco and Los angles.

In these systems, AI models are used to perform many tasks such as traffic sign recognition (TSR). TSR is used in autonomous cars so to model the driving behavior based on the signs encountered on the road.

These systems are so vulnerable to main Advesarial ML attacks, which aims to compromise the availability and the performance of the model, by creating adversarial samples injected in the used dataset, based on the goal of an attacker [1].

Among these attacks, poisoning attacks are one of the most important. These attacks acts during the training stage of a ML model and plans to poison the training dataset by injecting poisoned samples into it. This way, the model prediction at deployment time are compromised.

One of pioneer poisoning attack is BadNets which aims to add to clean images, some patches, obtaining the poisoned images. This way. whenever the model predicts on a poisoned image, it predicts a target label desired by the attacker.

One of the main drawbacks of these attacks is that they are so vulnerable to main dataset cleaning techniques. More specifically, in case of a dataset composed by few samples, a simple human check of these samples is enough to spot the poisoned samples among the clean ones, causing the attack to fail.

For these reasons, in this work we compare one of the latest few-shot classification model, LDC [2], with some non-few-shot S.O.T.A. and baseline models in the traffic sign recognition task.

III. Related Works

Among the main Few-shot learning models, CLIP is one of the most used. It is a multi-modal text-image, encoder

grounded on Contrastive Learning, which is able to generate text and image embeddings which are close to each other for text describing the provided image. CLIP demonstrated high transfer learning capabilities, allowing its usage for Zero-Shot Learning (ZSL) and Few-Shot Learning (FSL) tasks [3].

CLIP can be used in ZSL so to predict on novel classes. To do so, new text captions need to be created, one for each class. To make prediction on an unseen sample, the embedding of the sample and of the text caption, for each class, is generated using CLIP. Then, the cosine similarity between the sample and the caption embeddings is calculated as $\frac{z_x - z_{y_i}}{||z_x|| * ||z_{y_i}||}$, with z_{y_i} the embedding of the caption related to class y_i and z_x the embedding of the unseen sample. Finally, all the similarities are given to a softmax layer, obtaining a vector of probabilities s_x (logits). The predicted class is the one related to the maximum probability in the vector s_x. More specifically, the predicted class is $y = \arg\max_{y_i} s_x[y_i]$.

CLIP logits suffers of inter-class confusion. This means that CLIP used for classification, is not able to correctly distinguish classes which can be close to each other, generating logits which lead to misclassifications. This problem is due to the way in which CLIP is trained. This can limits the application of CLIP in FSL.

To overcome this problem, LDC has been proposed. It aims to learn the noise Δs which represents the difference between the clean logits $\hat{s}$ and the CLIP ones s^{ZS}. This way, the CLIP logits are improved by using multiple multi-level adapter fusion modules. These modules are able to transform the CLIP logits into the clean ones by learning the noise [2].

Among the non-few-shot models, there are many created ad hoc for solving traffic sign recognition task on specific datasets. The most of them are grounded on the usage of a Convolutional Neural Network (CNN) backbone to extract features from a given image and a classifier sub-network to make predictions based on the obtained embedding.

One of the first relevant model in this domain is based on a MCDNN proposed by Dan et al. This architecture is based on the usage of multiple DNN networks, of which predicted embeddings are averaged so to obtain the final prediction. A DNN plans to distort the image during the training phase, by applying multiple transformations such as rotation and scaling. This model was able to achieve 99.46% of accuracy on the GSTRB dataset. We refer to this model as *MCDNN*.

Another important model is the S.O.T.A. proposed by Mishra and Goyal [4] trained on the GTSRB dataset of german traffic sign. It is based on a CNN using four convolutional layers with ReLU and two max pooling layers with dropout so to prevent overfitting. The model was able to achieve a macro average F1 score of 0.99 and 99.76% of accuracy. We refer to this model as the *CNN*.

IV. Proposed Work

Our works aims to compare the performance of the LDC few-shot model and those of non-few-shot models by using a Paired T-test on multiple datasets.

The motivations under this experimentation are related to the effectiveness of some mitigation techniques performed on poisoning attacks in case of datasets composed by few samples. In fact, having few samples in the dataset, allows to easily mitigate these attacks via a manual check of the samples, discarding poisoned ones. This may lead in an increase of attack's failures and decrease of accidents regarding autonomous vehicles. This is an important point to consider, since these poisoning attacks, like BadNets [7], have proved high effectiveness on traffic sign recognition systems, acquiring 97% of success rate by poisoning only 2% of the training dataset [1].

So, by checking if there is no relevant difference between the performance of non-few-shot and few-shot models, we can use few-shot ones, being more robust to these attacks in traffic sign recognition applications.

A. T-test

The idea used to compare the few-shot model with non-few-shot models is grounded on the usage of the statistical Paired T-test in order to check if there is a significant difference in performance between the two considered models.

The T-test is used in order to check if there is a statistical difference between two means of two populations of independent samples. In ML, the T-test is used to compare the performance of ML models evaluated on multiple datasets. The Paired T-test requires that the models are trained on the same datasets and with the same evaluation technique.

The problem here is that we cannot use the same training dataset for both of the models, because the few-shot requires a specific configuration of it, with K samples for each of the N classes (N-way K-shot configuration).

For this reason we proposed a variation of the Paired T-test which is grounded on the same assumptions that the original version uses.

Our T-test works as follows:

- Choose K different and independent datasets on which evaluate the models
- For each dataset, we evaluate the models using a stratified *5-fold Cross Validation* (CV). The stratified methodology plans to split the folds such that, the portion of samples associated with each label is the same for each fold. The evaluation metric used in the macro average F1 score on the test fold.
- At each iteration of the CV, the test set is composed by using one fold as also the training set using the remaining 4 folds. In order to train the few-shot model in a N-way K-shot scenario, we extract another dataset from the training one, by picking K samples for each label at random, forming the training dataset used for the few-shot model.
- The test is still paired since both of the models are tested on the same test fold and generally evaluated on the same datasets.

At the end, we will obtain one F1 score for each dataset and for each of the two tested models. The T-test will so

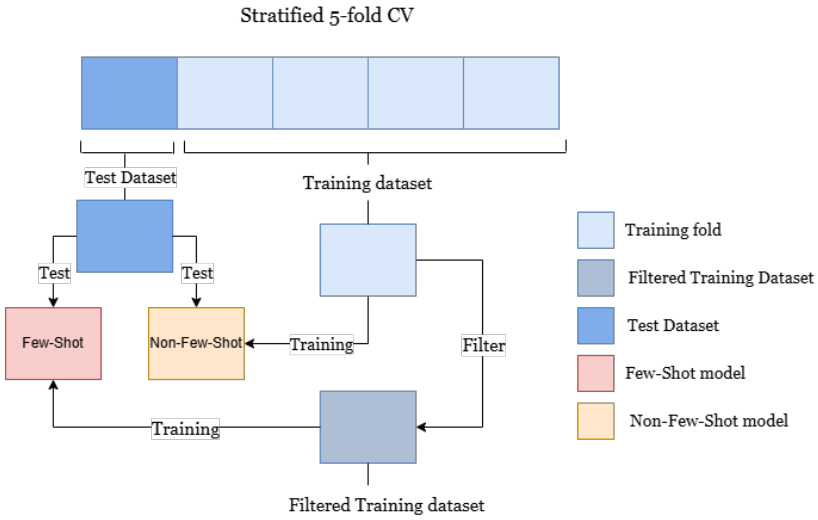

Fig. 1: Stratified 5-fold Cross Validation used during the Paired T-Test

check if there is a significant difference between the mean F1 scores of the two models.

To be sure that we have for each class at least K samples to use in the training dataset of the few-shot models, the original datasets used are filtered. More specifically, since the training dataset is composed by 4 folds and we are going to use the few-shot model with 16-shots, we need to have classes with at least 20 samples each, so to have at least 16 samples x class in the 4 folds of the training dataset.

The filter procedure plans to remove from the datasets, those classes which has a number of samples associated to them lower than 20.

The Cross-Validation strategy used in the paired T-test is represented in the Figure 1.

B. Models Used

The models used in this work are the CNN and MCDNN mentioned in the Related Work section. Also, some baseline models have been used.

These models work on samples that represent the HOG (Histogram of oriented gradients) features of the images. These features are created basing on the direction of the gradients of the image, calculated using Sobel kernels.

As baseline models, some pre-NN models are used. More specifically we used an *HOG SVM* model inspired by those built by Song et al. It plans to extract the HOG features from the given image and use them as input to the SVM model to conduct multi-class classification in a one-vs-all scenario. We refer to this model as the *HOG-SVM*.

Also, a simple KNN model is used to classify unseen traffic signs based the most frequent label among its nearest neighbors considering samples in the training dataset. The HOG features of the images have been used also in this case. We refer to this model as the *HOG-KNN*.

C. Datasets Used

To conduct the Paired T-test, the same datasets need to be used for both the tested models. In our experimentation we used only traffic sign recognition datasets, each of which contains signs coming from different countries of the world.

Those datasets have been found in the literature and on the main Machine Learning platforms like Kaggle and Hugging Face. Also, they have been implemented using the provided libraries by the same sources.

To perform the T-test using stratified 5-fold CV, we need to be sure to have at least 20 samples for each class, so to perform the N-way 16-shot few-shot training. To do so, datasets are filtered so that classes with less than 20 samples have been removed.

The datasets used are:

- *GTSRB*
  Dataset of German traffic signs. Used as benchmark in current traffic sign recognition systems. It has 43 classes and more than 50000 samples. After filtering, all the classes are preserved.
- *RTSD Cleaned*
  Dataset of Russian traffic signs. Obtained using an object detector on images in the RTSD original dataset, getting the cropped images of the signs. It has 198 classes and more than 104000 samples. After filtering, 59 classes have been removed and 103000 samples are still in the dataset.
- *Chinese Dataset*
  Dataset of Chinese traffic signs. It has 58 classes and more than 6000 samples. After filtering, 11 classes have been removed and 5900 samples are still in the dataset.
- *Bangladesh Dataset*
  Dataset of Bangladesh traffic signs. It has 43 classes and more than 70000 samples. After filtering, all the classes are preserved.

	CNN	*MCDNN*	*HOG-SVM*	*HOG-KNN*
8-shot 10 epochs LDC	False (1.12)	True (31.6)	True (73.5)	True (17.4)
16-shot 10 epochs LDC	False (-0.27)	True (25.0)	True (40.6)	True (13.9)
16-shot 30 epochs LDC	True (-17.1)	False (1.73)	False (0.87)	False (-0.6)

TABLE I: Results of the T-test. True indicates that there is a significant difference, while False indicates that there is no significant difference in the F1 means. For each cell, there is reported (a) with a the t-statistic. The t-critical is 2.44 for all the experiments.

	CNN	*MCDNN*	*HOG-SVM*	*HOG-KNN*
8-shot 10 epochs LDC	$0.06/0.1$	$0.06/0.95$	$0.06/0.91$	$0.06/0.85$
16-shot 10 epochs LDC	$0.11/0.1$	$0.11/0.95$	$0.11/0.91$	$0.11/0.85$
16-shot 30 epochs LDC	$0.88/0.1$	$0.88/0.95$	$0.88/0.91$	$0.88/0.85$

TABLE II: In this table there are reported the average F1 scores obtained, for each model, during the T-tests. Each cell a/b indicates that a is the average F1 score of the model on the row and b this of the model on the column

- *PTSD*
  Dataset of Persian traffic signs. It has 43 classes and more than 14000 samples. After filtering, all the classes are preserved.
- *Indian Dataset*
  Dataset of Indian traffic signs. It has 58 classes and more than 14000 samples. After filtering, all the classes are preserved.
- *BelgiumTS*
  Dataset of Belgian traffic signs. It has 62 classes and more than 7000 samples. After filtering, 10 classes have been removed and 6900 samples are still in the dataset.

V. Experiment Settings

All the experiments have been performed using two separate machines. *Machine 1* with i7-10750H and RTX 2060 mobile GPU. *Machine 2* with i7-13750H with RTX 4050 mobile GPU. We used Python as programming language and PyTorch and Scikit-learn as main libraries to implement the code.

For the Paried T-test we used a confidence value of $c = 0.05$. The few-shot model has been evaluated in N-way 16-shot and N-way 8-shot scenarios.

The hyperparameters used for the models are the following ones:

- *LDC*
  Epochs 10 and 30. Learning rate 0.0001
- *CNN*
  Epochs 10. Learning rate 0.0005
- *MCDNN*
  Epochs 10. Learning rate 0.001
- *HOG-SVM*
  Default Scikit-Learn hyperparameters have been used.
- *HOG-KNN*
  $K = 3$ has been used. So, 3 nearest neighbors are considered during the prediction.

The usage of 30 epochs for LDC has been the result of an hyperparameter tuning phase conducted by performing some executions of the model training. Also the used learning rate has been obtained by a conducting the same procedure.

To extract the HOG features, we used 9 bins with blocks of $8x8$. These features are extracted using the OpenCV library.

The LDC with 10 epochs and the *HOG-SVM* have been trained using Machine 2, while the LDC with 30 epochs, the *CNN* and the *MCDNN* have been trained on Machine 2.

VI. Results

Results obtained by performing the T-test between the few-shot LDC model and all the other non-few-shot models considered, are reported in tables: Table I and Table II.

Unfortunately, due to a lack of time, the experiments on the 8-shot 30 epochs have not been conducted.

The cumulative training execution times of the models, for all the datasets used in the T-test procedure, are reported in the following list:

- 30 epochs *LDC* training lasted for approximately 27 hours.
- 10 epochs *LDC* training lasted for approximately 10 hours.
- *HOG-SVM* training lasted for approximately 14 hours.
- *CNN* training lasted for approximately 10 hours.
- *MCDNN* training lasted for approximately 23 hours.

VII. Conclusion

As reported by the executed T-tests, the LDC model has been able to provide comparable results respect to those obtained by the baseline models used, which are HOG-SVM and HOG-KNN, and respect to the S.O.T.A. models used.

More specifically, *LDC* with 30 epochs was successful in equalizing the performance of all the non-few-shot models showing a t-statistic which was between $-z$ and z with z the t-critical. Only against the *MCDNN* model, the t-statistic was a little bit closer to the t-critical respect to all the other models. Also, even if there is no significant difference between the performance of *LDC* and *HOG-KNN*, the t-statistic shows that those of *LDC* are a little bit better. Furthermore, *LDC* with 30 epochs was able to achieve, in his favor, significant difference in performance respect to those of *CNN*.

On the other hand, *LDC* with 10 epochs, in both 8 and 16 shot, has failed all the tests, showing results a lot worse than those obtained by the baseline models. For these cases, the t-statistic was so much higher than the t-critical. Only

compared to the CNN model, the 10 epochs LDC was able to achieve comparable performance.

These tests showed that there is no statistical difference in the performance of the few-shot *LDC* model and all the others, included baseline and S.O.T.A. models. So, we can use *LDC* for traffic sign recognition, having results comparable with those obtained by other models, but being more robust to main poisoning adversarial attacks, resulting is a safer model.

REFERENCES

[1] V. S. Barletta, C. Catalano, M. Colucci, M. De Vincentiis and A. Piccinno, "Measuring the risk of evasion and poisoning attacks on a traffic sign recognition system," 2024 IEEE International Workshop on Technologies for Defense and Security (TechDefense), Naples, Italy, 2024, pp. 138-143,

[2] Shuo Li, Fang Liu, Zehua Hao, Xinyi Wang, Lingling Li, Xu Liu, Puhua Chen, & Wenping Ma. (2025). Logits DeConfusion with CLIP for Few-Shot Learning.

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, & Ilya Sutskever. (2021). Learning Transferable Visual Models From Natural Language Supervision.

[4] G. S. Mishra J, "An effective automatic traffic sign classification and recognition deep convolutional networks," Multimedia Tools and Applications, 2022.

[5] Yucong, Song & Shuqing, Guo. (2021). Traffic sign recognition based on HOG feature extraction. Journal of Measurements in Engineering. 9. 10.21595/jme.2021.22022.

[6] Luísa Shimabucoro, Timothy Hospedales, & Henry Gouk. (2023). Evaluating the Evaluators: Are Current Few-Shot Learning Benchmarks Fit for Purpose?.

[7] Tianyu Gu, Brendan Dolan-Gavitt, & Siddharth Garg. (2019). BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain.