Measuring AI model performance and domain shift in unlabelled image data from real scenarios

Noelia Vallez[3], Rosana Rodriguez-Bobada[1], Aubrey Dunne[*,2], Jose Luis Espinosa-Aranda[1], Oscar Deniz Suarez[3]

[1] *Ubotica Technologies, Ciudad Real, Spain*
[2] *Ubotica Technologies, Dublin, Ireland*
[3] *University of Castilla-La Mancha (UCLM), Ciudad Real, Spain*

When an Artificial Intelligence model runs in a real scenario, two situations are possible: 1) the data analysed follows the same distribution as the data used for model training and therefore the model performance is similar; or 2) the distribution of the new data is different, resulting in lower model performance. This is called "data/domain shift" and its measurement is desirable in order to reduce it. For example, for a model trained using images captured with high brightness, a change in the sensor may produce darker samples and make the model fail. To mitigate this problem, the sensor can be configured to obtain brighter images and thus reduce data shift. The simplest way to measure the shift is to compare metrics for the two data distributions. However, data captured in the real scenario is not labelled and an alternative is needed. In this work we propose using the Jensen-Shannon divergence score to measure the data shift. Results, obtained by using 5-fold cross-validation, show high correlation between the proposed metric and the accuracy (-0.81, -0.87 and -0.91) when test samples are modified for different brightness, sharpness and blur. The approach has applicability to autonomously measuring domain shift in Earth Observation data.

1 Introduction

Deep learning models are usually trained using a collection of labelled data, the acquisition of which can be difficult, and which can have bias in sample selection. Thus, the training set does not always represent the variability of the problem and there is a need to adapt the model to a new data distribution in order to increase its performance. In some cases, objects are not correctly identified or images are misclassified due to unusual contrast, brightness, blur, etc. Some deep learning-based systems discard low quality images to improve overall performance, necessitating image re-acquisition. If the acquisition process is carried out manually, the user can adjust some parameters and check if the resulting image has better quality. If not, the system will need to change a set of camera parameters automatically to improve the outcome [1].

Deep learning models are based on the assumption that training and test data are independent and distributed in the same way. Unfortunately, factors such as image acquisition equipment, light conditions or camera angles depend on the real scenario and can reduce the model performance. Similarly, the performance of the imaging system can change over time, either due to system degradation or to environmental changes, e.g., Earth Observation (EO) image quality may change over mission lifetime as the satellite altitude decays. The difference between the training data distribution and the distribution of the scenario where a model is deployed is called domain shift [2]. Thus, the higher the domain shift is, the lower is the generalisation to the new data. Domain shift can be measured by a drop in accuracy in the test set. However, when the model is finally deployed in the real scenario, such as on an EO satellite, new data coming from the sensor has not been labelled and the accuracy can not be obtained to measure if the model is obtaining good results. This is especially true of the new class of missions enabled by on-board Artificial Intelligence (AI), where insights are extracted from images directly on-board (for subsequent downlink) and the associated image data may never be downlinked. In this context, we propose an "unsupervised" metric to assess the domain shift existing between two datasets based on the divergence between training and real test distributions.

*Corresponding author. E-Mail: aubrey.dunne@ubotica.com

2 Methods and Materials

2.1 Dataset and Models

Rather than using an existing EO dataset for initial experimentation, a terrestrial proxy was chosen. This was done in order to access a large dataset of images that is captured from a range of different sensors, and contains a variety of imaging artefacts such as out-of-focus blur and overexposure. The selected proxy dataset is the "Diabetic Retinopathy Detection" Kaggle dataset [3], constituting colour retinal images captured with fundus cameras, and classified as either DR (classes 1-4, 9316 images) or Non DR (class 0, 25813 images). In order to minimise the loss of information when resizing the images for input to the CNN model, the images are pre-processed according to the "squaring" method described in [4]. All images were distributed across 5 subsets to perform a 5-fold cross validation (5fcv). With this distribution, 5 different models were trained. The CNN architecture employed during our evaluation is an InceptionV3 with an additional fully-connected layer of size 512. To simulate different acquisition conditions and induce domain shift, three types of image modifications with different parameters have been applied to each test set: (1) Gaussian Blur with variances {1, 3, 5, 7, 9}; (2) Brightness increase by adding {12.5, 25, 37.5, 50, 62.5, 75, 87.5, 100}; (3) Sharpness with values {1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2}.

2.2 Neuron Activation Patterns

Neuron Activation Patterns (NAPs), introduced in [5], are the neural outputs from a single specified network layer, and can be used to measure how the model output is supported by the information learnt from the training samples. Once a model is trained, the NAPs for close-to-output Neural Network layers can be obtained from all training samples that are correctly classified by it. Once obtained, these NAPs can be clustered into groups, with the Silhouette coefficient [6] measuring how cohesive and separated are the resulting clusters. For a given sample, the Silhouette score, s, is obtained as:

$$s = \frac{b - a}{max(a, b)} \tag{1}$$

where a is the mean distance between a sample and all other points in the same class, and b is the mean distance between a sample and all other points in the next nearest cluster. A value close to 1 indicates that the sample is far away from the neighbouring clusters, whereas a value close to 0 indicates that the sample is on or very close to the decision boundary between two neighbouring clusters. In addition, a value close to -1 may indicate that those samples have been assigned to the wrong cluster. Hence, the Silhouette score for a set of samples is given as the mean of all sample scores. This value will be higher when clusters are dense and well separated.

Figure 1 shows the Silhouette score obtained for some of the InceptionV3 layers in our model. It is possible to see how the score value increases towards the top of the CNN with the second-to-last layer having the highest value (notice that the last layer is excluded from the analysis because its output is the class and not a set of features). The Silhouette score can be used to choose the most appropriate layer for which to collect and monitor NAP divergence in the subsequent phase.

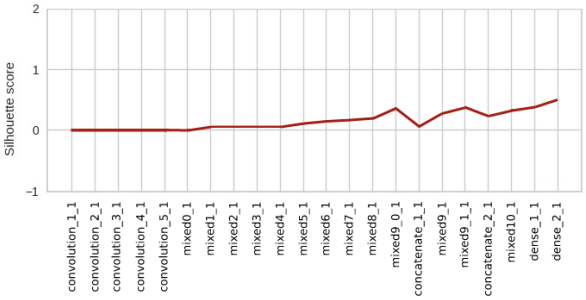


Figure 1: Silhouette score across InceptionV3 layers

2.3 Data Shift Metrics

We consider each neuron value from the NAP as a single random variable and therefore each of the 512 variables of the selected layer will follow a probability distribution. The distribution can be the same for both training and real scenario datasets when the shift between domains is close to 0, or can have different values depending on the divergence between training and real datasets. We have analysed three metrics for measuring the distribution shift between two distributions, training, Q, and modified test, P: Kullback-Leibler Divergence [7], Jensen-Shannon Divergence [8], and Fortiss NAP Monitoring [5].

2.3.1 Kullback-Leibler Divergence

The Kullback-Leibler (KL) divergence score (also known as relative entropy) [7], between two distributions P and Q is commonly expressed as:

$$KL(P \parallel Q) = \sum^{x \in X} P(x) \times log \frac{P(x)}{Q(x)} \tag{2}$$

The KL divergence is the sum of the probability of each event in P multiplied by the log of the probability

of the event in P over the probability of the event in Q. A score of 0 indicates that both distributions are equal. If not, the score is a positive number and the higher its value the larger is the difference between both distributions.

When the probability of a possible value is large in P but small in Q, the resulting divergence is large. On the contrary, when the probability of an event is smaller in P than in Q, the divergence is not as large.

Finally, if there is more than one variable, the global KL divergence can be calculated as the sum of all the individual KL scores. In this case we have obtained and summed the KL divergence score of each of the 512 NAP values.

2.3.2 Jensen-Shannon Divergence

The Jensen-Shannon (JS) divergence score is derived from the KL divergence but, contrary to KS, JS is symmetrical [8]. It is defined as:

$$JS(P \parallel Q) = \frac{1}{2} \times KL(P \parallel M) + \frac{1}{2} \times KL(Q \parallel M) \quad (3)$$

where M is calculated as:

$$M = \frac{1}{2} \times (P + Q) \quad (4)$$

2.3.3 NAP Runtime Monitoring

Monitoring the newly acquired NAPs is proposed in [5] to check if the model outputs are supported by the training data. When an unseen NAP pattern appears, it may indicate the presence of an out-of-distribution sample, i.e., samples that differ from the training in-distribution samples. Therefore, monitoring the presence or the absence of the runtime NAPs in the training NAP database can be used as a measure of the domain shift. Two scenarios can then be distinguished:

- The NAP pattern of the image can be found in the training patterns. The network is able to extract the features from the image and classify it accordingly.
- The NAP pattern is not found in the training patterns. This often happens when the input has features that have not been seen before and may indicate a domain shift.

In our case, the percentage of test samples with NAPs contained in the training NAP database is obtained (hereafter referred to as the NAP Runtime Monitoring (RTM) metric). The higher this percentage, the more similar both distributions are.

3 Results

Blur, Brightness and Sharpness have been applied to the test set of each of the 5fcv partitions. In all cases, classification accuracy, KL and JS divergences, and RTM metrics have been obtained.

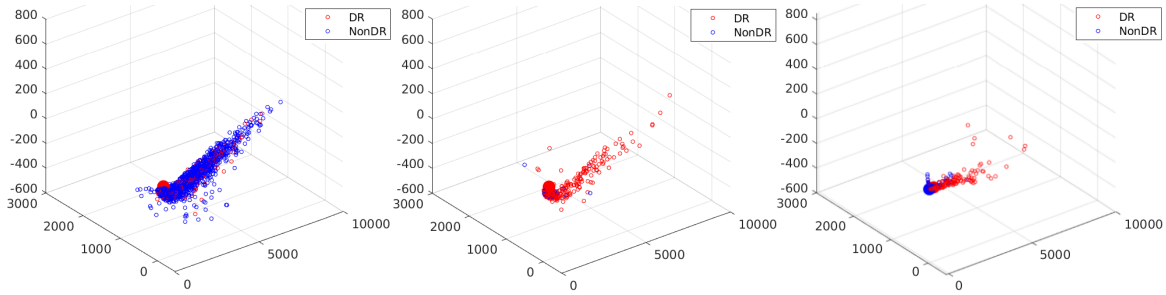
The training, test and modified test set NAP values have been plotted to see how the distribution varies when the modification is applied (Fig. 2). To achieve this, the selected 512 neuron activation values have been reduced to only 3 by applying Principal Component Analysis (PCA). In all cases the explained variability is over 99% which means that most of the information is retained in these three components and the plots are representative. When the test images are modified, the resulting NAP distribution is less similar to the training one (compare Fig. 2-b and Fig. 2-c with Fig. 2-a).

The correlation between the accuracy and each of the metrics was calculated using the Pearson Correlation Coefficient (PCC). A PCC of 0 means that changes in accuracy do not correlate with changes in the metric. A PCC close to 1 or -1 means that between both the accuracy and the metric variable values there is a strong relation. Table 1 contains the correlation results for each image modification, fold and metric used. Figure 3 shows the relation between accuracy and JS after blurring the images.

	Blur			Brightness			Sharpness		
Fold	JS	KL	RTM	JS	KL	RTM	JS	KL	RTM
Fold 0	-0.82	-0.81	0.98	-0.77	-0.91	0.97	-0.66	0.69	0.60
Fold 1	-0.88	-0.90	0.98	-0.97	-0.94	0.78	-0.97	0.87	0.05
Fold 2	-0.90	-0.99	0.27	-0.99	-0.95	-0.88	-0.87	-0.92	0.18
Fold 3	-0.97	-1.00	0.79	-0.99	-0.89	-0.07	-0.91	0.53	-0.03
Fold 4	-1.00	-0.87	0.86	-0.33	-0.23	-0.09	-0.96	0.87	0.72
Mean	−0.91	−0.91	0.78	−0.81	−0.79	0.14	−0.87	0.41	0.30
SD	0.07	0.08	0.29	0.28	0.31	0.75	0.13	0.76	0.34

Table 1: Correlation results measured using PCC. JS=Jensen-Shannon divergence, KL=Kullback-Leibler divergence, RTM=Runtime Monitoring.

The JS divergence score shows the strongest correlation average values (-0.91, -0.81 and -0.87) followed by the KL divergence score (-0.91, -0.79 and -0.41). The correlation between accuracy and {KL, JS} divergence scores is expected to be negative as the accuracy decreases when the divergence between training and test distributions increases. On the other hand, the correlation between the accuracy and the number of samples with NAPs in the training database should be positive since more patterns found means that the test images are more similar to the training samples. The difference between the JS and KL metrics may be caused by the symmetric property of JS. For KL



(a) Training distribution **(b)** Test distribution **(c)** Modified Test distribution

Figure 2: Comparison of the NAP distributions from training, test and modified sets.

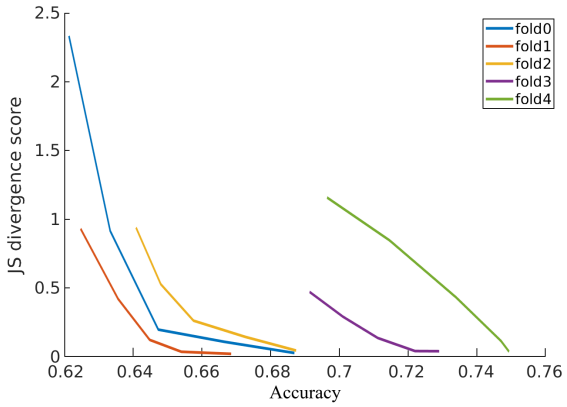


Figure 3: Accuracy and JS values after applying Blur. Accuracy decreases as JS divergence increases for all 5 folds.

divergence, when the probability for an event in P is large but the probability for the same event in Q is small, there is a large divergence. Conversely, when the probability in P is small and the probability in Q is large, there is also a divergence, but it is not as significant as in the first case. This asymmetry can lead to different results depending on the order of the distributions, which is not always desirable. This is particularly important in scenarios such as clustering, where the goal is to measure the similarity between distributions without biasing one over the other.

In contrast to KL divergence, JS divergence gives the same weight to a probability difference between the compared distributions, whether it increases or decreases, ensuring that the divergence measure is the same regardless of the order of the distributions being compared. In this context, an increase or decrease in the probability value of a certain event is equally important, as it measures the magnitude of the difference between both distributions.

The lowest results obtained by the RTM metric may be caused by the loss of information from binarising the NAP or by the fact that changes in the output of one of the neurons can produce a pattern that is no longer in the training database but it is still correctly classified. It is worth noting that both JS and KL metrics compare NAP distributions without requiring/considering the ground truth label.

4 Discussion

Measuring the domain shift between training and real scenario distributions is crucial for tuning the image acquisition sensor parameters to optimize model performance. The proposed metric can be used as a guide to how the selection of a certain value to configure the sensor is affecting the accuracy. The first step will be to select a set of samples from the real scenario with the default sensor parameters and calculate the domain shift metric between it and the training set. Then, a modification of the parameters is performed obtaining a new set of modified samples and its corresponding domain shift metric. If the value of the metric increases after the parameter modification, it indicates a larger divergence between the distributions and, therefore, a decrease in accuracy. On the other hand, if the divergence is smaller, the gap between the distributions shortens with the new value of the selected parameter and the accuracy of the model increases.

Although one of the advantages of the metric is that it can be used for unlabelled data, it is not capable of detecting class changes. For example, having two distributions with exactly the same NAP values, but with each of the NAPs corresponding to different classes depending on which distribution they belong to, will lead to a domain shift equal to 0 when in reality there is a huge drop in accuracy.

Finally, the metric can be used to identify samples that belong to a different distribution and can potentially be used to indicate that the NN model needs to be updated due to domain shift.

Future work will explore the JS metric applied to an AI model trained and operating on real EO datasets,

in order to investigate the sensitivity and effectiveness of domain shift detection in real EO imagery.

The effect of the real data dataset size needs to be explored, particularly when this dataset is much smaller than the training dataset. A small real dataset size may impact the divergence metrics. For example, the model might perform well on the training set but poorly on the real data due to overfitting, leading to misleading divergence measures. Similarly, smaller real datasets can lead to higher variability in the estimated distributions, affecting the reliability of the divergence metrics. In this respect, the KL divergence might be more sensitive to the discrepancies between training and test dataset sizes, especially if there are events with zero probabilities in the real data that are not zero in the training set. Contrarily, the JS divergence tends to be more robust in handling small datasets due to its symmetric and bounded nature. It smooths out the differences by averaging the distributions, which can provide a more stable measure of divergence.

5 Conclusions

In this work a metric to quantify the difference between two data distributions has been proposed for the identification of domain shift for an NN model. The metric is obtained using the NAPs obtained from the second-to-last layer. The layer selected is the one with the higher Silhouette score, which indicates how well the NAP values of the layer separates the samples between the classes. Results, obtained by using 5-fold cross-validation, show the higher metric/accuracy correlation using the Jensen-Shannon divergence metric (values -0.81, -0.87 and -0.91) when test image samples are modified using different brightness, sharpness and blur parameters. The proposed metric can be used to calibrate the image acquisition process in order to minimize the effect produced by the domain shift, or for alerting to the presence of domain shift.

References

1. Rajan, S., Ghosh, J. & Crawford, M. M. Exploiting class hierarchies for knowledge transfer in hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing* **44,** 3408–3417 (2006).
2. Pooch, E. H., Ballester, P. L. & Barros, R. C. Can we trust deep learning models diagnosis? The impact of domain shift in chest radiograph classification. *arXiv preprint arXiv:1909.01940* (2019).
3. *Diabetic Retinopathy Detection Kaggle Competition* `https://www.kaggle.com/c/diabetic-retinopathy-detection/overview`.
4. Fong, C. Analytical methods for squaring the disc. *arXiv preprint arXiv:1509.06344* (2015).
5. Cheng, C.-H., Nührenberg, G. & Yasuoka, H. *Runtime monitoring neuron activation patterns* in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (2019), 300–303.
6. Zhu, L., Ma, B. & Zhao, X. Clustering validity analysis based on silhouette coefficient [J]. *Journal of Computer Applications* **30,** 139–141 (2010).
7. Pérez-Cruz, F. *Kullback-Leibler divergence estimation of continuous distributions* in *2008 IEEE international symposium on information theory* (2008), 1666–1670.
8. Nielsen, F. On the Jensen–Shannon symmetrization of distances relying on abstract means. *Entropy* **21,** 485 (2019).