Using Open Data for Training Deep Learning Models: A Waste Identification Case Study

Juan Carlos Arbeláez[1](✉), Paola Vallejo[1], Marta Silvia Tabares[1], Jose Aguilar[1,2,3], David Ríos Zapata[1], Elizabeth Rendón Vélez[1], and Santiago Ruiz-Arenas[1]

[1] Universidad EAFIT, Medellín, Colombia
{jarbel16,pvallej3,mtabares,jlaguilarc,drioszap,erendonv,sruizare}@eafit.edu.co, aguilar@ula.ve

[2] Universidad de Los Andes, Merida, Venezuela

[3] IMDEA Network Institute, Madrid, Spain
jose.aguilar@imdea.org

Abstract. One of the main challenges of building commercial solutions with Supervised Deep Learning is the acquisition of large custom-labeled datasets. These large datasets usually fit neither commercial industries' production times nor budgets. The case study presents how to use Open Data with different features, distributions, and incomplete labels for training a tailored Deep Learning multi-label model for identifying waste materials, type of packaging, and product brand. We propose an architecture with a CBAM attention module, and a focal loss, for integrating multiple labels with incomplete data and unknown labels, and a novel training pipeline for exploiting specific target-domain features that allows training with multiple source domains. As a result, the proposed approach reached an average F1-macro-score of 86% trained only with 13% tailored data, which is 15% higher than a traditional approach. In conclusion, using pre-trained models and highly available labeled datasets reduces model development costs. However, it is still required to have target data that allows the model to learn specific target domain features.

Keywords: Deep Learning · Transfer Learning · Waste Identification · Solid Waste Management

1 Introduction

Deep Learning (DL) models are made of multiple layers of nonlinear functions that automatically learn features from raw inputs. This property of the DL models allows us to reduce the costs and time of feature engineering compared to developing Machine Learning (ML) models [1]. However, for the model to learn the mapping function between the raw inputs and the desired outputs and at the same time achieve good generalization, it is required to feed them with a large amount of labeled data (for supervised learning). The prevalence of large datasets is one of the main reasons for the success of DL models, especially the Convolutional Neuronal Networks (CNN) that have dominated the state-of-art in the later years in computer vision problems such as object detection, image classification, or semantic segmentation [16]. Nevertheless, gathering the required amount of labeled data for building strong solutions is impossible in many cases. This could be due to many reasons, for example, high costs of data labeling when experts' knowledge is required, for instance, in medical fields. Another case is when the observations (input data) are difficult to obtain in the case that rarely occurs, or there are privacy concerns for acquiring the data. For commercial solutions, the time required to assemble large datasets usually does not fit the market times or tight budgets used in the industry.

Particularly, one of the biggest challenges of the development of computer vision applications is the amount of data required [8]. Different methods are used to train DL models with scarce data: (i) Transfer Learning that aims to use knowledge of a source model trained on different but related domain and task [23]; (ii) Data augmentation that applies a set of transformations (i.e. Affine or point operators) to each sample of the training dataset to generate new ones [7]; (iii) Synthetically generated data that uses simulations to create new data (in the case of computer vision tasks, renders or animations are used for training the model [13]); and (iv) Few-shot learning that aims to train a model with fewer samples but generalize to unfamiliar samples [6]. In all of the presented methods, the main challenge is the discrepancy between the domains (source and target), where the feature space or label space is different or has different distributions.

Open Data (OD) refers to data that can be published and reused without price or permission barriers [12]. In the later years, the availability of large OD online has increased through platforms such as Kaggle or Roboflow that allows users to share labeled datasets. Although it is possible for academic purposes to use them for bench-marking, they can not be used directly for industrial or commercial purposes as they differ from the target domain and task. For instance, they have different labels, or the images are related but in different contexts. This article presents an approach for using OD from multiple domains and labels to train a supervised image multi-label model (Sect. 3). The approach is composed of three elements: a training pipeline (Sect. 3.1), an attention module (Sect. 3.2), and a Focal loss with incomplete data (Sect. 3.3). We validated the proposed approach in a case study of waste identification (Sect. 2), where the model is trained to predict the material, type of packaging, and product brand of waste in an image (Sect. 4). The proposed approach reached an average F1-macro-score of 86% on the target dataset and 85.8% evaluated in the source domains (Sect. 4.3). Section 5 present the research conclusions.

Figure 1 shows a random sample from the source (also includes 12.5% of target domain images) dataset with the three categories of labels. The label of the material, the type of packaging, and the product brand occupy the first, second, and third positions, respectively.

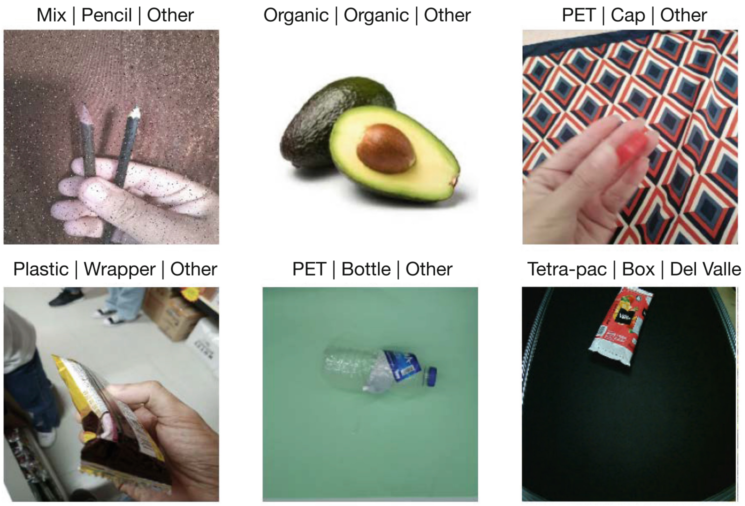
Mix | Pencil | Other
Organic | Organic | Other
PET | Cap | Other
Plastic | Wrapper | Other
PET | Bottle | Other
Tetra-pac | Box | Del Valle

Fig. 1. Random sample from the source (also includes 13% of target domain images) dataset with the three categories of labels. The label of the material, the type of packaging, and the product brand occupy the first, second, and third positions, respectively. Section 6 describes the datasets used and the link where they are available.

2 Waste Identification Challenge

The world's cities generate 1.5 billion tonnes of solid waste per year, and it is expected to increase to 2.2 billion tonnes by 2025 [4]. Therefore Solid Waste Management (SWM) is a global issue that affects every individual and government. At least 21% of municipal waste is not managed correctly, directly impacting public health and the environment [4]. Performing the correct waste separation increases the material recovery rate and reduces environmental impacts and economic expense [2]. Nevertheless, performing the waste separation is difficult as it is affected by multiple factors such as physical, human behaviors, knowledge, and policies, among others [14].

ML models are frequently used for waste classification and location in waste management systems. The CNNs are commonly used to perform these tasks [15,17,19,22] but two main barriers are typically mentioned: (i) The size of the datasets: given that the visual appearance of waste can vary a lot due to high deformations, dirt, and material degradation, it is required to have many samples of the same type of object that includes multiple variations. (ii) Location tailored: brands, products, objects' appearance, and even recycling categories vary from place to place, such that training datasets need to be custom for a specific location. These reasons prevent the reuse of datasets for different places.

In the proposed case study, a DL model is trained using datasets from different locations with different products, contexts, brands, and labels. The predictive function takes an image of a waste and outputs three categories of predictions (Fig. 1 shows a random sample from the source):

1. Material: one of 12 classes: plastic, PET, cardboard, aluminum, paper, Tetrapac, glass, steel, paper towel, mix, paper receipt, and organic.
2. Type of packaging: because of not only the material but the recycling process and type of the object gives insights about the recycling category. For instance, aluminum cans can be reused to contain food again, unlike aluminum used on other products. The model predicts 12 classes of packaging: wrapper, container, box, can, bottle, foil, cap, pencil, cutlery, organic, battery, and masks.
3. Product brand: Extended Producer Responsibility (EPR) are policies where the producer is responsible for the post-consumer stage of product life, including recycling [10]. Thus 39 local brands (from Colombia) are predicted by the model, and one additional to include "other" brands that are not taken into account but allow the model to not classify mandatory in one of the defined brands. Particularly, brand identification is the most local-tailored prediction category compared to material and type of packaging that share more similar features in different places.

The source dataset labels are composed of three categories: the material, the type of packaging, and the product brand occupy the first, second, and third positions, respectively.

3 Our Open Data Approach to Train Deep Learning Models

The presented approach aims to improve the performance of a model on a target domain that contains fewer samples (or none) required to learn a predictive function with samples of other domains. This problem is formally defined under the term of Transfer Learning [20] where a *Domain* $\mathcal{D}$ is described by two parts, a feature space $\mathcal{X}$ and a marginal probability distribution $P(\mathbf{X})$ where $\mathbf{X} = \{x_1, \ldots, x_n\}$. The feature space $\mathcal{X}$ comprises all possible features, and $\mathbf{X}$ is a particular set of the domain with n number of instances. For a domain $D = \{\mathcal{X}, P(\mathbf{X})\}$. A *task* $\mathcal{T}$ is defined by two components as well, a label space $\mathcal{Y}$ and a predictive function $f(\cdot)$ trained from the feature-label x_i, y_i where $x_i \in \mathbf{X}$ and $y_i \in \mathbf{Y}$. $\mathbf{Y} = \{y_1, ..., y_n\}$ are the corresponding labels of the particular learning sample set $\mathbf{X}$, and the function $f(\cdot)$ is the predictive function that can be seen as $P(y|x)$, the probability of y given a feature x. In a general case, we have two domains with their related tasks, the *source domain* $\mathcal{D}^{\mathcal{S}} = \{\mathcal{X}^{\mathcal{S}}, P(\mathbf{X^S})\}$ and its respective task $\mathcal{T}^{\mathcal{S}} = \{\mathcal{Y}^{\mathcal{S}}, P(\mathbf{Y^S}|\mathbf{X^S})\}$. Similarly, the *target domain* $\mathcal{D}^{\mathcal{T}} = \{\mathcal{X}^{\mathcal{T}}, P(\mathbf{X^T})\}$ with $\mathcal{T}^{\mathcal{T}} = \{\mathcal{Y}^{\mathcal{T}}, P(\mathbf{Y^T}|\mathbf{X^T})\}$. Therefore, TL is defined in this context as the process of improving the target predictive function $P(\mathbf{Y^T}|\mathbf{X^T})$ using information from $\mathcal{D}^{\mathcal{S}}$ and $\mathcal{T}^{\mathcal{S}}$ with the condition that $\mathcal{D}^{\mathcal{S}} \neq \mathcal{D}^{\mathcal{T}}$ or $\mathcal{T}^{\mathcal{S}} \neq \mathcal{T}^{\mathcal{T}}$.

For our case, we have a source composed of multiple domains, each one with its related task: $\mathbf{D^S} = \left[(\mathcal{D}^{\mathcal{S}}, \mathcal{T}^{\mathcal{S}})_1, \ldots, (\mathcal{D}^{\mathcal{S}}, \mathcal{T}^{\mathcal{S}})_n\right]$ for improving one target predictive function in the domain $\mathcal{D}^{\mathcal{T}} = \{\mathcal{X}^{\mathcal{T}}, P(\mathbf{X^T})$.

The proposed approach comprises three elements: a training pipeline, an attention module placed in the head of a pre-trained *feature extractor*, and a

Focal loss for incomplete data. The approach is defined for a multi-label task where the labels can be grouped into categories that are not mutually exclusive. The same approach can be used for traditional multi-label or classification problems.

3.1 Training Pipeline

The pipeline is performed in three stages. The first is to prepare the datasets, then model training, and the last is the model evaluation.

Data preparation. After gathering the source datasets, the first step is performing a *label mapping*. In the *label mapping*, each class of the target categories (i.e., material or brand) is labeled encoded (each class is represented by a consecutive integer number starting by zero), and an additional class *"Unknown"* represented by -1 is added. The mapping is performed as follows: if the class is in the set of target classes, then assign it the corresponding label (y_i). Otherwise, it is assigned to *"Unknown"*:

$$y_i = \begin{cases} y_i^T, & \text{if } y_i^S \in \mathcal{Y}^\mathcal{S} \\ -1, & \text{otherwise} \end{cases}$$

The *label mapping* allows easy processing of all the samples of the source datasets without the need to manually relabel, and at the same time, at the training time, to detect if a sample has an *"unknown"* label (if its value is less than 0). Later, the labels are one-hot encoded by category; each category label is represented by a K dimensional vector corresponding with length of the number of classes of the category, and one element is equal 1 that corresponds to the position of the label encoded and the rest remains 0. The final label is a vector of the concatenated one-hot encoded categories of $\mathbb{R}^{\sum K_i}$. Figure 2 shows an example of the encoded label process for each image, where each category generates a one-hot encoded vector concatenated to produce the final label. If the label is *"Unknown"*, a zero vector of K dimension is generated.

Model training. The model training is performed with Adam [9], a stochastic gradient-based optimization. The train split is composed of all source datasets randomly shuffled where a training sample is a pair feature, label (x_i, y_i). The training is performed in two steps:

1. Attention training: The parameters of the attention module described in Sect. 3.2 are tuned first, the rest of the model parameters are frozen (not updated during backpropagation), and the attention module is only trained with the target dataset on one prediction category (i.e., brand or material). The category selection is based on the one that can provide more local or domain-tailored information (i.e., the brand category).
2. Prediction training: After the attention module is trained, the layers in charge of the final prediction are trained. Both the attention module and the *feature extractor* are frozen during this step, and the model is trained with the source datasets.

The proposed pipeline intends that the *feature extractor* is already pre-trained in a general dataset (ImageNet) and "knows" to extract features from images. The attention module is trained to weigh features relevant to the tailored dataset for the target task. Therefore, step 2 (Prediction Training) later pays more attention to these features during the training of the final prediction layers. The model is trained using backpropagation with the loss presented in Sect. 3.3.

Model evaluation. To evaluate the model with highly unbalanced datasets with missing labels, a non-zero average macro F1-score is used as a performance metric. The F1-score is the harmonic mean between the precision and recall, thus penalizing if the model's prediction is biased to a majoritarian class. The average macro F1-score is calculated by each class and averaged by the prediction categories. Given that some datasets could not be present in some classes, it is not considered in the average.

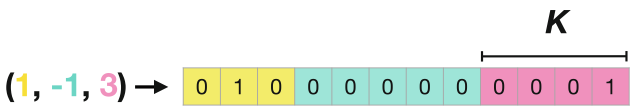

Fig. 2. Example of the encoded label process for each image. For each category is generated a one-hot encode vector that is concatenated to produce the final label.

3.2 Attention Module

The model architecture is composed of a *feature extractor*, an attention module, and a custom head for performing the predictions (Fig. 3).

The *Feature extractor* is a CNN model pre-trained on ImageNet where the prediction head is removed and is only used in the last convolution layer output. For the feature extraction, any state-of-art model could be used, such as VGG16 [18], MobileNet [5], or ResNet50 [3]. For example, ResNet50 (achieved the highest score) produces a *feature map* of $(7, 7, 2048)$ dimensions (red in Fig. 3). Resnet50 is a CNN architecture that uses skipped connections to fix the gradient-vanish problem related to backpropagation on deep models. The *feature map* is passed by a CBAM Attention module [21] that is composed of channel (green on Fig. 3) and spatial (yellow on Fig. 3) modules.

(i) The attention module produces a channel attention map (M_c) using the *feature map* (F) channels inter-relationship, such that each channel can be considered a feature detector. The channel attention emphasizes what is meaningful in an input image. In the channel module, the spatial dimension is squeezed by adding two pooling operations over the channel dimension that produces an average (F^c_{avg}) and max (F^s_{max}) pooled features and passed by the same Fully Connected (FC) network:

$$M_c(F) = \sigma(FC(F^c_{avg} + FC(F^c_{max})))$$

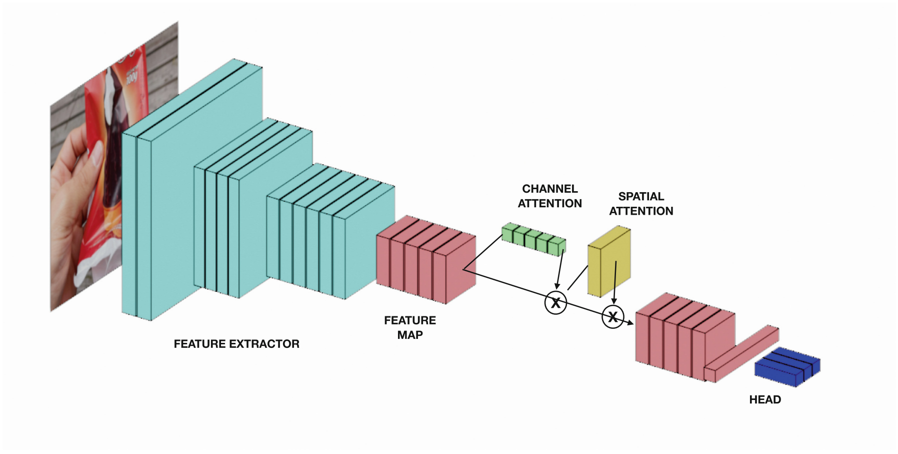

Fig. 3. Model architecture composed of a Feature extractor (in blue) that produces a feature map (red). The feature map is passed by a CBAM Attention with spacial (yellow) and channel (green) modules, and finally, fully connected layers for the prediction for each category. (Color figure online)

(ii) The spatial attention module produces a map (M_s) using the features' relationship. It determines the relevant information of the feature map (F) and encodes where to emphasize or decrease its contribution to the prediction. The spatial attention module is calculated by performing a *Sigmoid activation* (σ) of the convolution ($f^{n\times n}$) over the concatenated a max (F^s_{max}) and average pooled-features (F_{avg}) of the *feature map*.

$$M_s(F) = \sigma\left(f^{n\times n}(F^s_{avg}; F^s_{max})\right)$$

Each one of the attention maps multiplies the *feature map*; the channel attention vector multiplies each channel of the map and the spatial each feature (Fig. 3). The attention module (spatial and channel) are trained only on one prediction category with the target dataset. Later, their weights are frozen, and the rest of the model is trained for all the categories.

3.3 Focal Loss for Incomplete Data

The Focal Loss [11] is used for training models with highly unbalanced datasets. For instance, on object detection usually there is more background than foreground objects, which causes the model to be biased for the amount of one class. For dealing with dataset unbalance, there are other techniques, such as dataset sampling, which can be performed by oversampling the classes with fewer observations, or undersampling the majoritarian classes. Another technique commonly used is to add a weight to each sample depending on its class that defines their contribution to the loss during training (Weighted loss). However, both of these techniques usually do not work well when there are large differences between the classes (i.e., Brands distribution in Fig. 6) because training is inefficient and the model tends to over-fit given that a small set of samples are continually repeated.

Focal Loss (FL) is intended to work when there is an extreme class unbalance by adding a term $(1-p)^\gamma$ to the Cross-Entropy loss that allows reducing the contribution of easy samples focus on the difficult ones. The parameter γ adjust how much easy samples are weighted down:

$$FL(p_i) = -(1-p_i)^\gamma \log(p_i)$$

For our case, the multi-label problem can be decomposed as a multiple classification problem, given that each category is independent. We have a composed loss L of each categories loss L_c weighted by factor:

$$L = \sum_{1}^{c} W_c L_c$$

Each category loss is computed using the Focal Loss, and each sample (i) is only considered if the label of its category is known. Given that the "unknown" labels are set to a zero-vector and the Cross-Entropy loss is computed by multiplying the log of the predictions by its true probability (t_i), the loss for each category is computed as follows:

$$L_c = -\frac{1}{n_c} \sum_{i}^{n} (1-p_i^c)^\gamma \log(p_i^c) t_i^c$$

Each category loss is normalized by the number of known samples (n_c), where p_i^c is the model prediction of the sample i regarding the category c and t_i^c is the true label of the sample regarding the same category.

4 Definition of a Multi-label Deep Learning Model Using Our Open Data Approach

4.1 Dataset Acquisition and Preparation

In our case study, we train a predictive function that takes an image of waste and outputs three categories of predictions: its material, type of packaging, and product brand (Sect. 2). For the acquisition of the target data, a photography device (A in Fig. 4) was designed to control environment variables such as lighting or background, and perform an efficient data collection. The photography device is composed of 5 elements (B in Fig. 4).

1. Chasis in an aluminum profile that allows configuring the components in multiple positions; 2. Profile unions; 3. Multiple cameras that can be positioned with a support that allows to place them in any position and rotation; 4. Configurable lighting ring of three colors (natural, warm, and cold) and ten light intensities; 5. Background panels that can be changed to use different colors.

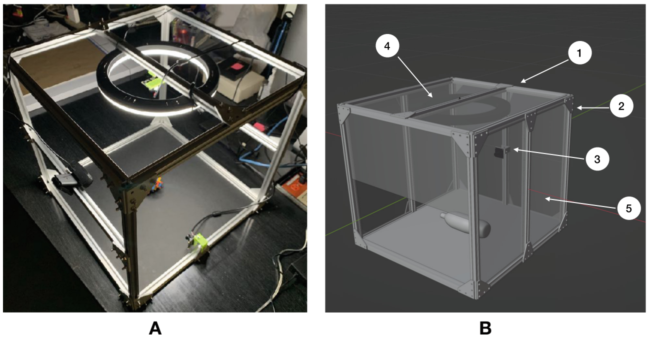

Fig. 4. **(A)** Photography device used for the target dataset acquisition. The photo shows three cameras and a lighting ring. **(B)** Photography device design, composed of five main parts: Chasis in aluminum profile, profile unions, multiple cameras, lighting ring, Background panels

The target dataset comprises 624 products commonly consumed in Universidad EAFIT - Colombia. An ID is assigned to each product, labeled according to the three categories to predict (material, type of packaging, and product brand). Generic products are also included for products that are not possible to know any labels. For each product, multiple photos are taken with three deformations: (0) No deformation is applied to the product, (1) Mild deformation: the product is opened for consumption and has some deformation, and (2) Severe deformation is applied to the packaging. Additionally to the photos taken with the device, 191 photos of products in different contexts were taken. Figure 5 shows random samples from the target dataset.

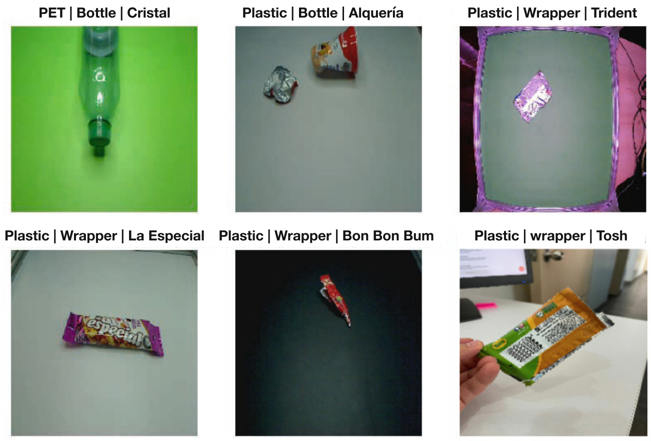
PET | Bottle | Cristal
Plastic | Bottle | Alquería
Plastic | Wrapper | Trident
Plastic | Wrapper | La Especial
Plastic | Wrapper | Bon Bon Bum
Plastic | wrapper | Tosh

Fig. 5. Random sample images from the target dataset, taken with the photography device and with random context

The target dataset is composed of 11.207 images, from which 25% is used for testing. Figure 6 shows the distribution of the target dataset of the three prediction categories.

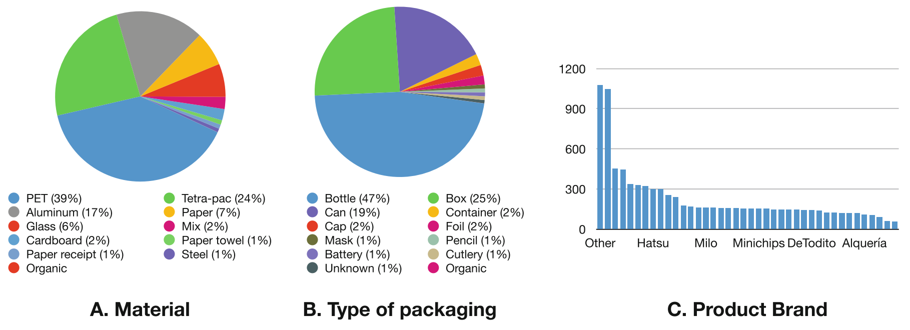

Fig. 6. Class distribution of target dataset of the 3 distribution categories

The source dataset has been created from 34 online datasets and contains 75.101 images, from which 20% is used for testing. In the next link are available the sources, the class contribution to the source dataset, and its respective references: Open Data waste datasets.

4.2 Model Training

The training dataset split comprises the source and target training splits. First, the attention module (Sect. 3.2) is trained for brand prediction as it contains more target-tailored information. After the attention module is trained, the rest of the model (type of packaging and material) is trained, freezing the weights of the attention module. Finally, the model training is performed with Adam [9] with the loss function described in Sect. 3.3. All the models were trained for 18 Epochs in total, ten epochs for the attention module and eight epochs with Early-stopping for the rest of the prediction heads(material and type of packaging).

Five models were trained, four of them using the proposed approach with different *feature extractors*: **(i) OpenData-VGG6** using VGG16 [18] as feature extractor, **(ii) OpenData-MOBIL** using MobileNet [5] as feature extractor, **(iii) OpenData-RESN** using ResNet50 [3] as feature extractor, and one **OpenData-NoFL** with ResNet50 as feature extractor and trained without Focal Loss; **(iv) BASE**, the traditional approach where a ResNet pre-trained model is fine-tuned with Cross Entropy loss.

Figure 7 shows the training curves of the highest performance model (OpenData-RESN). There is a larger difference between validation and training split in the brand category training because it is more difficult to make generalizations as it contains more categories and many of them look very similar in general (the package) but with some details different (the logo of the brand). This is less the case in material and packaging type training, where, for example, for the packaging type, the object shape characteristics are more universal.

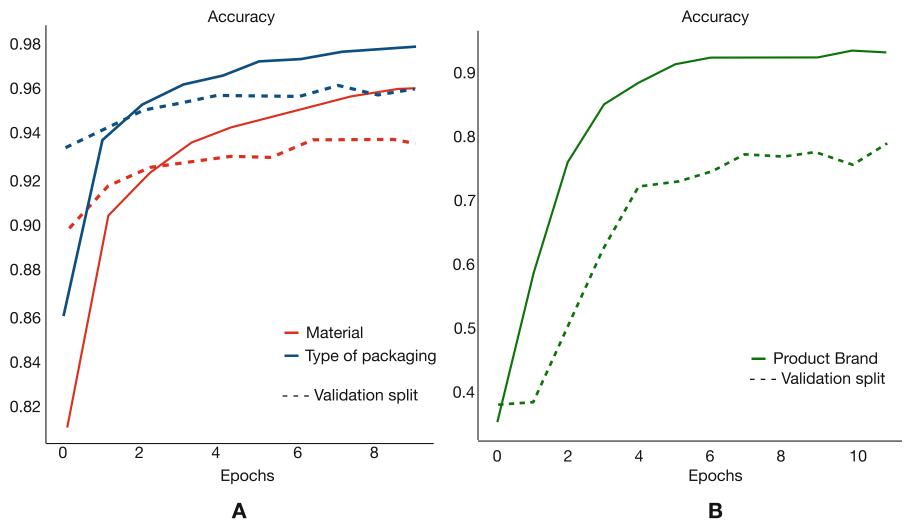

Fig. 7. Training curves of the highest performance model (OpenData-RESN). In **A**, the training of material and type of packaging model, and in **B**, Brand and attention module training. In dashed, the performance of the model in the validation split (5% of training dataset).

The data pre-processing was the same for each of the models evaluated, the images were loaded and converted from RGB to BGR, then each color channel was zero-centered for the ImageNet dataset without pixel intensity scaling. After the data pre-processing, feature extraction was performed using a pre-trained CNN model directly from the images without performing features selection. Hyperparameter selection was manually performed on the cross-validation split (5% of training split), and the samples of each split remained the same in all the experiments (Table 1).

Table 1. Training hyperparameters used in all the experiments.

Hyperparameter	Value
Learning Rate	0.003
Epochs	10, 8
Batch size	500
Radom seed	1

4.3 Model Evaluation

Table 2 summarizes the performance of the evaluated models. In blue is the performance over the source dataset, and in green is the performance over the target dataset. The metric used is the non-zero average macro-F1-score. The non-zero means that classes without samples will not be considered in the F1-score computation to avoid division by zero. Given the unbalanced target dataset, the F1-score is used as the evaluation metric as it penalizes lower precision and recall. The proposed approach OpenData-ResN achieved the highest result with ResNet50 as the feature extractor. Although the average macro F1-score is almost similar between the sources and the target datasets (1%), there is a major difference in two categories: material (-15 on target) and brand (+23 on target) The difference in the material performance may be due to the prediction depends of specific visual features that can be miss leading, for instance, packages that may look like other material (plastic as aluminum). On the other hand, in source datasets, many samples are labeled as "other" that may look very similar to a brand in the target dataset.

Table 2. Models performance evaluation results. The metric used is the macro F1-score (Sect. 3.1)

	Source Dataset				Target Dataset			
MODEL	**F1 MAT**	**F1 TYPE**	**F1 BRAND**	**F1 AVG**	**F1 MAT**	**F1 TYPE**	**F1 BRAND**	**F1 AVG**
OpenData-RESN	**0,94**	**0,96**	**0,65**	**0,85**	**0,79**	**0,90**	**0,88**	**0,86**
OpenData-NoFL	0,94	0,95	0,65	0,85	0,74	0,84	0,88	0,82
OpenData-MOBIL	0,93	0,94	0,68	0,85	0,73	0,83	0,85	0,80
OpenData-VGG16	0,93	0,94	0,66	0,84	0,72	0,81	0,88	0,80
BASE	0,75	0,78	0,84	0,79	0,75	0,78	0,59	0,71

Figure 8 shows the confusion matrices of material and type of packaging of OpenData-RESN in the target dataset. Worth noting that the target dataset is extremely unbalanced (Fig. 6). Additionally, there is one class (organic) that does not have evaluation samples, thus, it is not taken into account to calculate the evaluation metric (non-zero average macro-F1 score) due to zero division in the recall computation. Also, most of the miss classification occurs in the classes with more samples ("easy classes"), and the reason for this could be the use of the Focal loss and that the model can learn the difficult classes due to it being trained in a larger domain.

In order to evaluate the effect of the Focal loss, two brand classifiers were trained with and without Focal Loss, achieving 84% with Focal loss and 59% trained with Cross-Entropy loss.

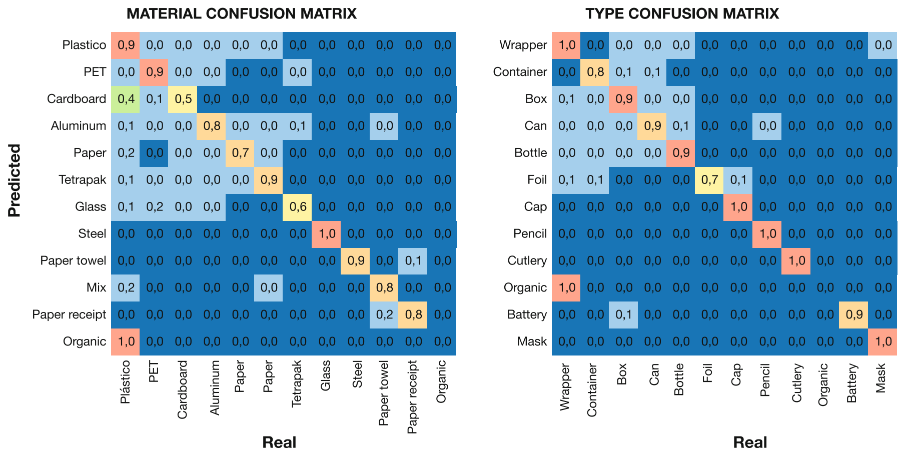

Fig. 8. Normalized confusion matrices by prediction of materials (left) and type of packaging (right) of the highest performance model (OpenData-RESN) on the target dataset.

5 Conclusions

This article presents a TL approach for using multiple source datasets from different domains with incomplete labels (Sect. 3) with three main elements: a training pipeline, an attention module, and a focal loss for missing labels. The approach was validated in a case study for waste identification using an image-based multi-label model with three categories: material, type of packaging, and product brand (Sect. 4).

The proposed approach performs better than the standard form of training custom models (15% increase in average macro F1-score, see Table 2) in the target domain. At the same time, the model trained with the proposed approach has a similar performance considering all the domains (85% average macro F1-score). The model was not biased toward the classes with more samples, given that the dataset was highly unbalanced (Fig. 6) and was able to differentiate the classes in both domains (Table 2).

The selection of the *feature extractor* has little impact on the performance regarding (1% in all the domains and 6% with the target). These differences could be due to the size of the feature map of each of the evaluated architectures.

The difference in the performance of the model regarding the prediction categories (11% between material and type of packaging) could be due to some categories using mode "generalizable" image features, for instance, the type of packaging is related to the shape of the object that is the same in different contexts opposite to the brand or the material that depends uniquely of the object appearance.

The inclusion of the Focal Loss has a positive impact on the performance of the models, 4% higher in the OpenData models on the target dataset (see

Table 2) and 25% higher in the brand classifier with Focal loss (see the last paragraph of Sect. 4.3).

Future work should focus on techniques and algorithms for using fewer samples of the target dataset and reducing the impact of the image background on inter-domain models. Additionally, exploring different electromagnetic spectra and lighting conditions with proper camera calibration to improve material identification.

6 Dataset Statement

This research uses 34 datasets collected from different sources. In the next link are available the sources and their respective references: Open Data waste datasets.

References

1. Arnold, L., Rebecchi, S., Chevallier, S., Paugam-Moisy, H.: An introduction to deep learning. In: European Symposium on Artificial Neural Networks (ESANN) (2011)
2. Chen, G., et al.: Environmental, energy, and economic analysis of integrated treatment of municipal solid waste and sewage sludge: a case study in China. Sci. Total Environ. **647**, 1433–1443 (2019)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
4. Hoornweg, D., Bhada-Tata, P.: What a waste: a global review of solid waste management (2012)
5. Howard, A.G., et al.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
6. Jadon, S.: An overview of deep learning architectures in few-shot learning domain. arXiv preprint arXiv:2008.06365 (2020)
7. Kaur, P., Khehra, B.S., Mavi, E.B.S.: Data augmentation for object detection: a review. In: 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 537–543. IEEE (2021)
8. Khan, A.A., Laghari, A.A., Awan, S.A.: Machine learning in computer vision: a review. EAI Endorsed Trans. Scalable Inf. Syst. **8**(32), e4 (2021)
9. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
10. Leal Filho, W., et al.: An overview of the problems posed by plastic products and the role of extended producer responsibility in Europe. J. Clean. Prod. **214**, 550–558 (2019)
11. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
12. Murray-Rust, P.: Open data in science. Nat. Precedings, p. 1 (2008)
13. Nikolenko, S.I.: Synthetic data for deep learning. arXiv preprint arXiv:1909.11512 (2019)

14. Oluwadipe, S., Garelick, H., McCarthy, S., Purchase, D.: A critical review of household recycling barriers in the United Kingdom. Waste Manag. Res. **40**(7), 905–918 (2022)
15. Panwar, H., et al.: AquaVision: automating the detection of waste in water bodies using deep transfer learning. Case Stud. Chem. Environ. Eng. **2**, 100026 (2020)
16. Patel, R., Patel, S.: A comprehensive study of applying convolutional neural network for computer vision. Int. J. Adv. Sci. Technol. **6**(6), 2161–2174 (2020)
17. Qin, L.W., et al.: Precision measurement for industry 4.0 standards towards solid waste classification through enhanced imaging sensors and deep learning model. Wirel. Commun. Mob. Comput. **2021**, 1–10 (2021)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
19. Singh, S., Gautam, J., Rawat, S., Gupta, V., Kumar, G., Verma, L.P.: Evaluation of transfer learning based deep learning architectures for waste classification. In: 2021 4th International Symposium on Advanced Electrical and Communication Technologies (ISAECT), pp. 01–07. IEEE (2021)
20. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. J. Big data **3**(1), 1–40 (2016)
21. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
22. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1492–1500 (2017)
23. Zhuang, F., et al.: A comprehensive survey on transfer learning. Proc. IEEE **109**(1), 43–76 (2020)