JAISCR, 2023, Vol. 13, No. 1, pp. 75 – 92
10.2478/jaiscr-2023-0005

TRAINING CNN CLASSIFIERS SOLELY ON WEBLY DATA

Dominik Lewy, Jacek Mańdziuk*

Faculty of Mathematics and Information Science,
Warsaw University of Technology,
Koszykowa 75, 00-662 Warsaw, Poland

**E-mail: mandziuk@mini.pw.edu.pl*

Submitted: 24th March 2022; Accepted: 19th October 2022

Abstract

Real life applications of deep learning (DL) are often limited by the lack of expert labeled data required to effectively train DL models. Creation of such data usually requires substantial amount of time for manual categorization, which is costly and is considered to be one of the major impediments in development of DL methods in many areas. This work proposes a classification approach which completely removes the need for costly expert labeled data and utilizes noisy web data created by the users who are not subject matter experts. The experiments are performed with two well-known Convolutional Neural Network (CNN) architectures: VGG16 and ResNet50 trained on three randomly collected Instagram-based sets of images from three distinct domains: metropolitan cities, popular food and common objects - the last two sets were compiled by the authors and made freely available to the research community. The dataset containing common objects is a webly counterpart of PascalVOC2007 set. It is demonstrated that despite significant amount of label noise in the training data, application of proposed approach paired with standard training CNN protocol leads to high classification accuracy on representative data in all three above-mentioned domains. Additionally, two straightforward procedures of automatic cleaning of the data, before its use in the training process, are proposed. Apparently, data cleaning does not lead to improvement of results which suggests that the presence of noise in webly data is actually helpful in learning meaningful and robust class representations. Manual inspection of a subset of web-based test data shows that labels assigned to many images are ambiguous even for humans. It is our conclusion that for the datasets and CNN architectures used in this paper, in case of training with webly data, a major factor contributing to the final classification accuracy is representativeness of test data rather than application of data cleaning procedures.

Keywords: Classification, webly data, InstaFood1M, InstaCities1M, InstaPascal2M

1 Introduction

In recent years applications of deep learning to computer vision have moved the field substantially towards human-level performance. A great example of this trend are the results of ILSVRC (ImageNet Large Scale Visual Recognition Challenge) that plummeted from more than 25% top-5 error in 2010-2011, when shallow methods were used, to less than 5% in 2015 with the use of deep learning (DL). For reference, this result is below human level accuracy error of 5.1% [40]. This advancement was possible mainly due to large publicly available annotated datasets like ImageNet [40] or COCO [29]. Creation of such datasets requires substantial amount of time for manual categorization which is costly and is considered to be one of the greatest impediments in DL models development in many areas.

DL efficiently tackles a great number of computer vision tasks like image categorization [24, 43, 47, 17], object detection/localization [36, 37, 42], image segmentation [16, 41, 12], or image captioning [3].

1.1 Main contribution

In this work we experimentally verify the efficacy of an approach to training Convolutional Neural Network (CNN) classifiers on noisy web images annotated by the users (presumably their authors) who, in many cases, do not assign appropriate labels (annotations), as they are not obliged to follow any particular set of labeling rules. In popular non-expert domains (such as city or nature landmarks, food, monuments, cars, etc.) the availability of such loosely-tagged images in the Internet is abundant. Hence it is interesting and potentially promising to verify to which extent relying solely on webly data in the training process may still lead to high-quality classifiers. In summary, the main contribution of this work is threefold:

– Demonstrating the effectiveness of a standard approach to training a CNN image classifier with no use of expert labeled data or human expertise of any kind, based solely on non-expert labeled data (downloaded from the web users accounts) with high amount of label noise. The underlying claim is that CNN classifiers despite being trained on noisy-labeled data may accomplish meaningful classification accuracy when tested on class-representative samples.

– Creation of two webly datasets: a food-related one (*InstaFood1M* [26]), composed of 1 million images (10 categories, each with 100 000 samples) and the other one, depicting common objects (*InstaPascal2M* [27]), composed of 2 million images (20 categories, each with 100 000 samples). Both datasets were created by downloading images from Instagram. The latter one is a webly counterpart of PascalVOC2007 dataset [10]. Both sets are freely available for research purposes.

– Experimentally proving the efficacy of proposed approach in three distinct domains: landmark photos of metropolitan cities, images of popular dishes and images presenting common objects.

1.2 Related literature

Despite certain approaches to classification with scarce availability of expert labeled data, like *One-shot Learning/Few-shot Learning*, e.g. [2, 50, 11], synthetic data generation [8, 15, 20, 53, 45], *Transfer Learning* [54], or using expert labeled data to guide the learning process [48, 18], the problem of training deep neural networks in the case of a complete lack of expert labeled samples is not a common research topic.

Our approach relies on the use of large quantities of data downloaded from the internet. Clearly, the usage of webly data in DNN training is not a new concept. For instance, in [32] web data is utilized in the CNN training, though augmented with a certain category supervision process. Another class of algorithms called Multiple Instance Learning [56, 31, 9] relies on weakly-supervised learning, with labels assigned not to individual images but to groups of images. Yet another work assumes existence of some “easy images” (i.e. characteristic and well-framed images on a light background) [7], which are used to pre-train the network before noisy images appear.

Some methods use a mixture of expert-labeled and noisy samples in the training process, which is generally the most common scenario of utilizing web data in classification tasks, e.g. [51, 33, 4, 49, 52, 5]. Another way of web data utilization is its application as a means of model pre-training [30, 46, 21].

Two streams of research explicitly addressing the presence of noise in the data refer to the outliers handling by either filtering or label purification [1, 5, 55], and decreasing the outliers impact on the training process by adjusting the weights [28, 34], respectively.

The method proposed in this paper differs from the above-cited accomplishments by taking randomly collected web data *as is*, with no further manual modification of any kind. The webly data is used directly in the training process without employing auxiliary expert labeled images or specific training setting.

The two papers closest to our research are [19, 23] which also use webly data for CNN training. Our work differs from [19] in that in [19] a robust linear regression algorithm is trained on top of the features extracted by a CNN, while our method uses a fully trainable CNN for both visual feature generation and classification. In [23] the authors take a *curriculum learning* approach and enhance webly data training by using Google queries, which are considered to be “easy images” [7].

Our method refers to the idea of *Transfer Learning* (TL) [54], which has recently become a common aspect of the vast majority of deep CNN applications. TL is often realized with the help of one of the well-know large CNNs pre-trained on a large dataset (*ImageNet* [40] or *COCO* [29]). Such a pre-trained deep CNN is used as a starting point for almost any image related problem.

In the view of non-expert knowledge approach proposed in the paper, besides initialization based on *ImageNet*, we have also tested CNN training with random weight initialization to make sure that the proposed approach is capable of extracting meaningful patterns and does not simply rely on features developed during *ImageNet* pre-training. While both scenarios differ in the speed of learning (*ImageNet* based initialization visibly shifts up the starting accuracy), **in terms of ultimate accuracy, the advantage of domain-based initialization compared to learning from scratch is minimal** (section 3.7.2 presents the details).

One of the datasets used in our experiments (*Instacities1M* [13]) was previously utilized in the work related to embedding images in text specific vectors, like Glove or word2vec [14]. In [14] those embeddings were used to enhance the image retrieval quality. Our research objectives and proposed solution methods are clearly non overlapping with [14]. The other two datasets (*InstaFood1M* [26] and *InstaPascal2M* [27]) were prepared by the authors of the paper and this research marks their first use.

The remainder of this paper is arranged as follows. Section 2 presents proposed solution in more detail. The next Section describes three datasets used in the experiments, as well as experiment setup and technical details of the training procedure. Section 4 summarizes the results in terms of classification accuracy in the context of the quality of the input data labeling and representativeness of test images. Additional experiments aimed at automatic cleaning of the data (with no human assistance) are presented in Section 5. Conclusions and directions for future work are discussed in the last Section.

2 Proposed approach

Our goal is to develop high accuracy classifiers trained solely on photo images collected from web pages of randomly selected Internet users. This type of data includes relatively high percentage of non-representative images (e.g. a photo of a city park or a sand beach) and dubious or erroneous labels (e.g. a selfie with a cat in an apartment which is labeled as “New York city”).

On a general note the proposed approach can be characterized as follows (the details are provided in Section 3):

- Automated data collection by means of downloading images from random web pages based on respective category hashtags.
- Automated data cleaning with no need of human expert involvement.
- Using a pre-trained CNN architecture in the training process on the above noisy web data.
- Accuracy assessment on class-representative images.

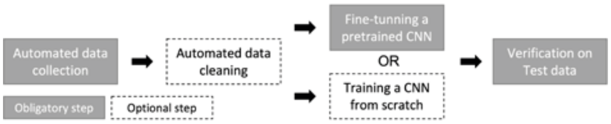

Automated data collection
Automated data cleaning
Fine-tunning a pretrained CNN
OR
Training a CNN from scratch
Verification on Test data
Obligatory step
Optional step

Figure 1. Flowchart of the proposed approach, with obligatory and optional steps indicated.

The proposed approach is summarized in Figure 1. Please note that a cleaning procedure listed in step 2 is optional and apparently, as discussed in Section 5, its usage does not improve the accuracy. This observation is one of the main conclusions of this work - in the case of webly data, the final classification quality is much more dependent on the representativeness of the test dataset than application of cleaning procedures.

Figure 2. Logos of the three Instagram-based datasets used in the experimental evaluation: *InstaCities1M* [13], *InstaFood1M* [26] and *InstaPascal2M* [27]. The last two sets were compiled by the authors.

Please also note, that although the use of pre-trained models in step 3) speeds up training, in terms of accuracy the learning from scratch leads to only slightly inferior results. Consequently, the use of networks initialized based on domain knowledge saves training time but does not increase the ultimate classification accuracy in a meaningful way. The details are presented in Section 3.7.

2.1 Motivation

The above-described problem setting addresses real business needs. For instance, many companies may require automated image categorization (e.g. automatic tagging of company's internal images) or an automated detection of a certain object (e.g. a safety helmet on construction site, safety gloves for ironwork or people presence in restricted areas) on the photos taken, for instance, by the CCTV camera. In many cases a relevant training data (ready to use) is not available or its availability is seriously limited.

Traditional shallow methods work with little data but they do not meet harsh accuracy requirements of commercial solutions. Deep learning methods could meet those expectations but need more training data which is costly to gather and in some cases hinders wider adoption of those algorithms in commercial solutions. The approach proposed in the paper relies on using large amounts of randomly collected web data (photos of required objects) and despite obvious flaws in this data (ambiguous or erroneous labels or non-representative images) alleviates the problem of scarce availability of human-labeled samples and enables wider adoption of DL methods in certain domains.

3 Experiment setup

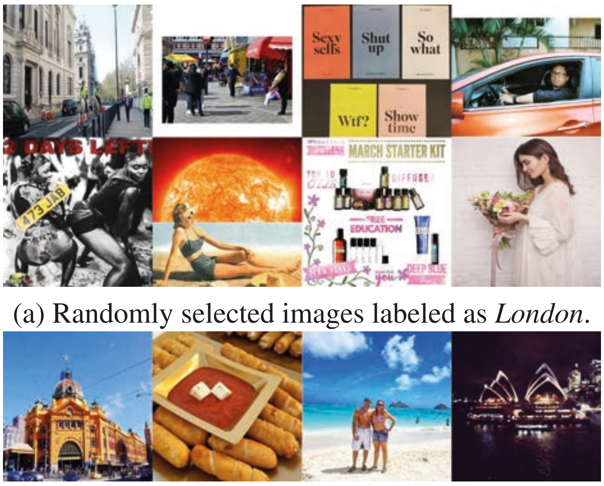
This Section presents detailed specification of three training datasets and two CNN pre-trained architectures used in the experiments along with technical details of the training procedure.

3.1 *InstaCities1M* set and *Clean* city-related test set

InstaCities1M [13] contains 1 million 224x224 colour images taken (presumably) in the following 10 cities: New York, Singapore, Miami, Toronto, Los Angeles, San Francisco, Melbourne, Sydney, London and Chicago. The data is divided into a training set (800 000 images), a validation set (50 000 images) and a test set (150 000 images).

InstaCities1M was created by downloading images tagged by a city name from Instagram. As Instagram primary reason is private sharing of images with other platform users, neither the images nor their descriptions are validated by experts of any kind.

Figure 3a shows that many of *InstaCities1M* images are not representative for *any* city (e.g. *a tattoo on a hand*) and even a human would have a hard job with correct classification of the majority of them. A manual inspection of 10 randomly sampled sets, each composed of 8 images, revealed that, on average, there was only 1 image per set that was truly class-specific. The remaining 7 presented a common city content which potentially might have been taken in some other cities, as well. This observation supports the claim about high level of noise in the dataset. Ambiguous classification and uncharacteristic images are just one source of the problems. Additionally, labeling is sometimes incorrect because people assign a city name different from the location in which the image was actually taken. Some examples of incorrect labeling are presented in Figure 3b. None of such dubious or incorrectly-labeled samples were manually removed from the dataset, as our aim is to propose and evaluate a fully automated approach to data collection and CNN training.



(a) Randomly selected images labeled as *London*.

(b) Examples of images with incorrect or unjustifiable labels.

Figure 3. *InstaCities1M* - The majority of (randomly chosen) images in Figure 3a are clearly not representative for London. In Figure 3b the images (from left to right) are assigned to Sydney, Miami, London and Singapore, resp. while the real classes are Melbourne (Flinders Street Railway Station image), undefined (there are no characteristic landmarks that could be helpful in identifying the location), undefined (but rather not London) and Sydney (Opera House image), resp.

In order to develop a test set composed of, most probably, properly labeled images we have additionally downloaded images from official Instagram accounts of the above-listed cities, except for Miami which seems not to have such an official account[1]. The following accounts were used for photo collection: nycgov (398 images), visit_singapore (1117), seetorontonow (1097), losangeles_city (1583), onlyinsf (1470), cityofmelbourne (1641), sydney (2968), london (8450) and chicago (2725).

100 randomly sampled subsets of these images (henceforth denoted as *Clean Random*) were created. Furthermore, in order to estimate the accuracy upper-bound a single set of the *easy to predict* images (*Clean Selected*) was defined in the following manner. For each class 300 images with the highest class probability returned by the trained network were selected regardless of the prediction correctness. Each of 100 instances of *Clean Random* as well as the *Clean Selected* set included 2700 images (300 per class). As stated above the underlying idea was to use this data as an independent test set composed of representative images.

3.2 *InstaFood1M* set and *Clean* food-related test set

InstaFood1M dataset [26] was prepared by the authors following the *InstaCities1M* structure. The set was created by downloading images from Instagram identified by particular hashtags. It contains 1 million 224x224 colour images from 10 following categories: Apple Pie, Burger, Donuts, French Fries, Hot Dog, Mac & Cheese, Pancake, Pizza, Spaghetti, and Steak, which constitute the top-10 food in the USA[2,3]. Analogously to *InstaCities1M* the data is divided into a training set (800 000 images), a validation set (50 000 images) and a test set (150 000 images).

Additionally, an independent test set with food images labeled by the experts was created as a subset of *food-101* dataset from kaggle.com[4], originally described in [6]. *food-101* contains 1000 images per each of its 101 classes. For the sake of direct comparison with cities classification experiments, the data was randomly down-sampled in each category from 1000 to 300 images, comprising the *Clean* food-related data set composed of 3000 images.

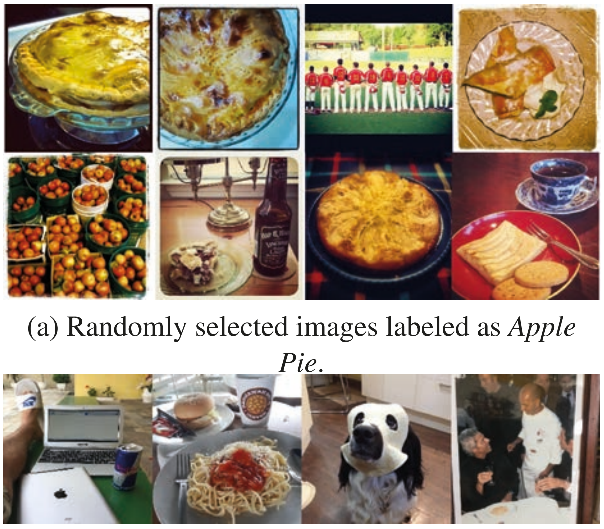
InstaFood1M suffers from a similar noise problem as *InstaCities1M*, albeit to a lesser extent. In the case of food, there are fewer non-representative images. Similarly to the previous case we had manually verified the content of 10 randomly sampled sets, each composed of 8 images, and observed that, on average, 5 out of 8 images were indeed class-specific.

[1] Consequently the experiments were finally performed with the remaining 9 cities.

[2] https://visual.ly/community/infographic/food/top-10-americas-favorite-foods

[3] https://food.ndtv.com/food-drinks/10-american-foods-777850

[4] https://www.kaggle.com/dansbecker/food-101



(a) Randomly selected images labeled as *Apple Pie*.

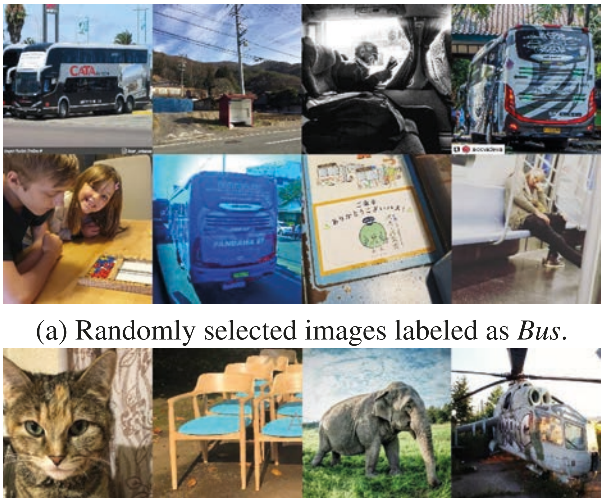
(b) Examples of dubious food-related images.

Figure 4. *InstaFood1M* - Some of the images presented in Figure 4a are clearly not those of apple pies though may be somehow linked to this category - e.g. a photo of apples. In Figure 4b the images (from left to right) are assigned to Apple Pie, Burger, Pancake, and Spaghetti, resp., although the real/appropriate classes are disputable. The first picture presents Apple laptop, the next one is a photo of spaghetti (first plan) and burger (in the background), the third one is a funny photo of a dog's face - in some sense resembling a pancake, and the last one was taken in a restaurant - probably with spaghetti on a small plate in the very bottom of the figure.

Generally, there are two sources of noise as presented in Figure 4b. The first one is the same as in the case of *InstaCities1M* - the label may not represent the content (e.g. Apple computer under Apple Pie hashtag). The other problem is more specific for this dataset as there can be more than one food category presented in the image (a typical example is a Burger with French Fries). Regarding the independent *Clean* test set compiled from *food-101* data, due to its manual labeling and verification, no incorrectly labeled or irrelevant images are expected. The issue which may still exist here is the co-appearance of more than one food category in an image.

3.3 *InstaPascal2M* set and *Clean* object-related test set

InstaPascal2M dataset [27] was also prepared by the authors, as a webly counterpart of PascalVOC2007 dataset [10]. The same 20 categories as in PascalVOC2007 were considered: Aeroplane, Bicycle, Bird, Boat, Bottle, Bus, Car, Cat, Chair, Cow, Dining Table, Dog, Horse, Motorbike, Person, Potted Plant, Sheep, Sofa, Train, TV/Monitor. In total 2.1 million 224x224 colour images were downloaded to comprise *InstaPascal2M*, further divided into a training set (1 600 000 images) a validation set (400 000 images) and a test set (100 000 images). The same sources of noise as in the two above-described datasets (incorrect labeling and ambiguous content) can be observed in *InstaPascal2M*, as depicted in Figure 5.



(a) Randomly selected images labeled as *Bus*.

(b) Some wrongly-labeled examples of images from *Car*, *Sofa*, *Dog* and *Airplane* classes.

Figure 5. *InstaPascal2M* - 4 out of 8 randomly selected images presented in Figure 5a actually depict a bus (3 pictures of real buses and a hand-drawing of a bus). Arguably another one (3rd, top row) could have been taken in a bus interior. Some of the remaining ones do not present a bus, but may possibly be linked to this category (a bus stop, or an interior of a metro cart - another means of transportation). In Figure 5b the images present a cat, a row of chairs, an elephant and a helicopter. The first two are wrongly classified and the last two are out of the scope of predefined categories.

As representative object-related (*Clean*) test samples, the images from test part of PascalVOC2007 were considered. In order to adjust PascalVOC2007 samples, which were originally meant for multi-label classification two metrics were calculated: accuracy on images that had only one class assigned (1905 samples), referred to as *accuarcy_filtered*, and accuracy on all PascalVOC2007 test images (4952). In the former case, which represents a typical multi-class setting, the accuracy calculation was straightforward. In the latter case, for each image it was checked whether the predicted class is on the list of assigned classes. This measure will be referred to as *accuarcy_one*.

3.4 Pre-trained CNN

Two CNN architectures pre-trained on *ImageNet* [40] are used as a starting point of the training procedure: VGG16 [43] and ResNet50 [17] (presented in Figs. 6 and 7, resp.). Standard VGG16 is extended by the dropout layers that we have added (highlighted in orange) to prevent over-fitting. Both architectures have 10 or 20 output neurons, depending on the number of classes in the three considered datasets (not 1000 as in the original versions). These two popular and quite different CNN architectures were selected to check generality of the proposed training approach and its independence of a particular pre-trained CNN selection.

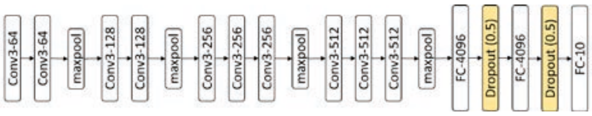


Figure 6. VGG16 architecture with 10 output neurons.

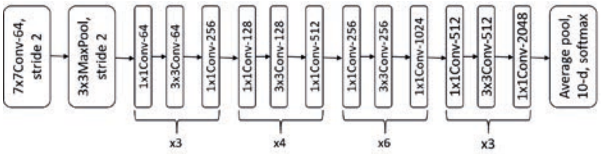


Figure 7. ResNet50 architecture with 10 output neurons.

3.5 Randomly Initialized CNN

For the majority of experiments we have used CNNs pre-trained on *ImageNet* as we would like to speed up the process and reduce the carbon footprint [44] of our experiments. Moreover, it is the best practice to use knowledge already available and build up on it. Nevertheless, we also wanted to demonstrate that high accuracy of the networks truly depends on the information available in the webly data and not on the features derived from *ImageNet* that were already available as a starting point. To this end, additional experiments with all three *webly* sets with randomly initialized ResNet50 network were conducted. For random initialization the weights were initialized as in [17] (i.e. sampled from Gaussian distribution with zero mean and standard deviation depending on the number of layers in the network) and the training was performed from scratch. The use of ResNet50 was motivated by its slightly higher performance and simpler training procedure.

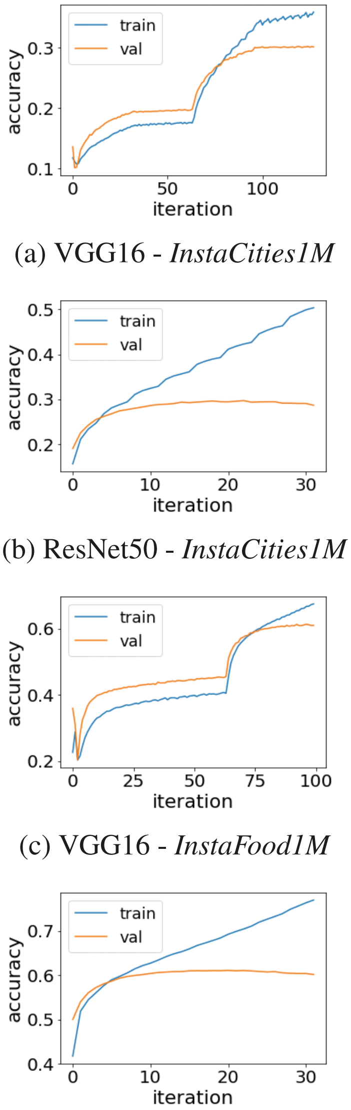
3.6 System parameterization

Initially four sets of experiments were run, each involving one of the two smaller training sets (*InstaCities1M* and *InstaFood1M*) and one of the two CNN architectures (VGG16 and ResNet50). VGG16 was extended by regularization in a form of a dropout after each fully connected layer. ResNet50 was used with no modifications, except for adjusting the size of the output layer (which concerned both architectures). Except for the above mentioned modifications both VGG16 and ResNet50 followed the original implementations described in [43] and [17], respectively.

Depending on the dataset and CNN architecture the learning process encompassed between 1 and 4 steps, each of them composed of 8 epochs (full training passes). Each epoch was further divided into 4 iterations for the reasons of error reporting (4 times per epoch). In the case of VGG16, the learning rate was set to $1E-4$ in the first step and decreased by the factor of 10 at the beginning of each subsequent step. Furthermore, training in the first two steps was limited to the last 3 layers only, with the remaining part being frozen. In subsequent steps the whole network was trained (albeit with lower starting learning rates). This strategy (in the case of VGG16) proved to yield better accuracy than training all weights from the beginning. In the case of ResNet50, the whole network was trained right from the start with the initial learning rate equal to $1E-5$. For both networks, when the accuracy started to plateau learning rate was decreased by a factor of 10. A 50 000-image validation set was used to prevent over-fitting. The batch size was equal to 64 images. In all experiments Adam optimizer [22, 39] was used.

Figure 8 presents example learning curves for both datasets. In the case of VGG16 a 4-step training process was performed. All layers were unfrozen after the second step, which caused a spike in both training and validation accuracy curves after iteration 64 for both datasets. In the case of

ResNet50 one-step training was sufficient and all weights were trained right from the beginning.



(a) VGG16 - *InstaCities1M*

(b) ResNet50 - *InstaCities1M*

(c) VGG16 - *InstaFood1M*

(d) ResNet50 - *InstaFood1M*

Figure 8. Training (blue line) and validation (red line) accuracy curves for both datasets.

3.7 Training and testing protocols

3.7.1 Main experiments

For both *InstaCities1M* and *InstaFood1M* sets the experiments were performed as follows[5]. First, 30000 images from each set were randomly selected and set aside to be used as a noisy (*Webly*) test set. After that both architectures were trained on the remaining 770000 images with the accuracy monitored based on 50000 validation images to prevent over-fitting. Each of the 4 experiments was conducted 3 times.

Additionally, two other data sets (one per each problem domain), described in Sections 3.1 and 3.2, respectively were used as *Clean* test sets, presumably without noise.

In the case of *InstaPascal2M* the experiments were performed on both noisy Instagram images and the representative ones. As noisy images 100 000 Instagram test samples left aside at the beginning of the experiment were used. These observations were divided into 10 disjoint sets so as to resemble the testing settings of the experiments with the two other datasets (*InstaCities1M* and *InstaFood1M*). As a class representative *Clean* set the test part of PascalVOC2007, described in Section 3.3, was used.

The main experimental hypothesis was that a classifier trained on (very) noisy Internet data could still provide high quality predictions on representative data.

3.7.2 Additional experiments

Additional experiments conducted on all three *webly* data sets aimed at verifying the importance of *ImageNet* initialization in the training process. Furthermore, we tested whether using higher volumes of webly data would lead to performance increase. The experiments were limited to ResNet50 architecture as it offered slightly higher accuracy with simpler training procedure. For each *webly* data set the following 4 experiments were performed. First, the ResNet50 network with random initialization was trained based on half of the training and validation data. Then, analogous training was performed with the same network initialized on *ImageNet*. Both experiments will be referred to as *small webly*. Afterwards, the same experiments were repeated using the entire *webly* data for training / validation subsets - referred to as *big webly* experiments (the proposed approach is visualized for a sample class in Figure 9). In both *small webly* and *big webly* runs the whole *Webly* test set (composed of 30 000 samples for *InstaFood1M* and *InstaCities1M*, and 100 000 samples for *InstaPascal2M*)

[5] The source code for all experiments is available at https://github.com/SzefKuchni/Insta_codes

was used to check the accuracy of the trained models. Each of the 4 experiments was repeated 3 times.

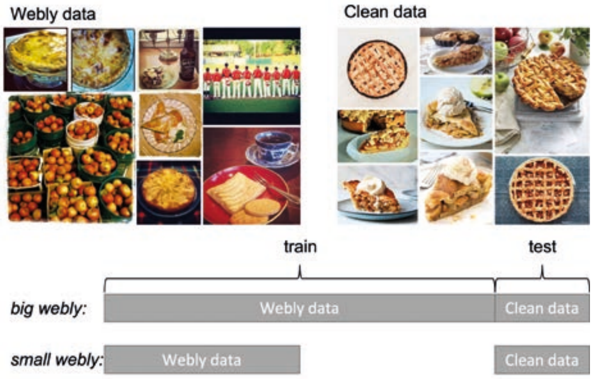


Figure 9. The figure illustrates the baseline idea of the proposed approach. In the upper part, two types of datasets for "apple pie" category are visualized: webly data that contains some erroneous images (due to noise) and clean dataset that contains only class-representative images. In the lower part, the concept of the experiments is visually explained by means of indicating the dataset types and their sizes.

4 Experimental results

4.1 Training on webly data

On *InstaFood1M* the results achieved with VGG16 and ResNet50 are close to each other, as presented in Table 1. For both architectures test accuracy on user Instagram images (denoted as *Webly* in the table) is lower than on manually labeled data from kaggle.com (denoted as *Clean*). Test results on *Clean* data are repeatable and of high accuracy (up to 89.1% on average), exceeding *Webly* test results by up to 24.5 p.p. Lowering the amount of training data (*small webly* experiments) causes only slight accuracy drop on both *Clean* and *Webly* test sets. Similar trends can be observed for *InstaCities1M* (Table 2) where tests on *Clean* set collected from official Instagram cities accounts yield higher accuracy than tests on *Webly* data from Instagram user accounts, albeit the accuracy is generally lower compared to food data. For randomly selected images from the *Clean* test set (*Clean Random* in Table 2), depending on the architecture, the accuracy is up to 14.1 p.p. higher than for noisy images (*Webly*). The difference raises significantly (up to 50.5 p.p.) for the selected *easy to predict* data (*Clean Selected*, defined in Section 3.7.1). In the case of *InstaCities1M* the impact of training set size is the highest among the three sets as reducing the amount of training data by 50% (*small webly*) caused an accuracy drop of 5.7 p.p. for *Clean Random* and 7 p.p. for *Clean Selected*, respectively.

The results for *InstaPascal2M* are presented in Table 3. The mean accuracy on *Clean* dataset (PascalVOC2007) when the whole training data was used (*big webly*) reached 83.8% and 80.5% for $acc\_filtered$ and acc_one measures, respectively. The results for *Webly* test data were on average 31.9 p.p. worse. Reducing the training data (*small webly*) caused a relatively minor accuracy deterioration (between 2.1 and 3.3 p.p.).

4.2 Training on clean data

For the sake of establishing the reference point we performed experiments aimed at verifying the level of accuracy which can be achieved using exclusively *clean* (class representative) training data.

In *food-101* set, the same 3 000 images (which composed the *Clean* test dataset) were used for testing and the remaining 7 000 for training (6 000) and validation (1 000), resp. In the case of PascalVOC2007, we followed the standard data split suggested by the authors of this set, i.e. for training and validation we used 5011 images and for testing 4952 images. In the case of cities-related data, despite efforts, we could not reach any meaningful results, due to small number of samples (2700) and their relatively lower specificity.

The results of training with *clean* data did not match the accuracy achieved with *webly* data training in none of the two domains in which we had enough *clean* data available. For food-related data the accuracy presented in Table 1 (*clean*) is on average a few percent worse than using *big webly* training data. The same observation is valid in common objects related data, presented in Table 3, where *clean* training leads to worse performance than training on *big webly* data for both measures described.

4.3 Training with/without pre-training

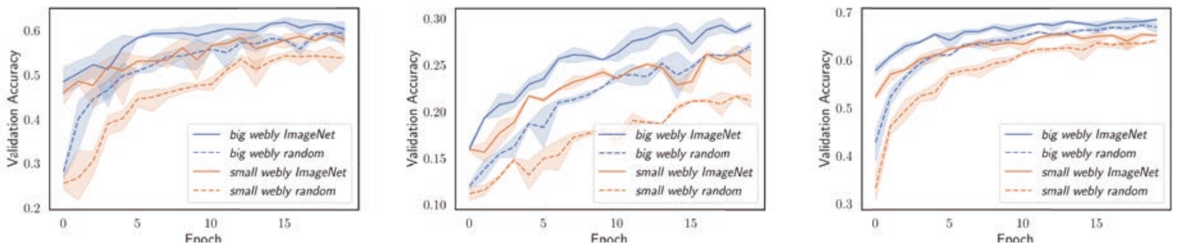
Learning curves aggregating results of 4 experiments, each with 3 runs, with *webly* training data sets are presented in Figure 10. It can be seen in the figures that indeed providing more data increases

Table 1. *InstaFood1M*. Accuracy results on *Webly* and *Clean* test sets described in Section 3.7.1. *big webly* and *small webly* refer to utilization of the entire training set and half of this set, resp. (cf. Section 3.7.2). *clean* training data refers to class representative images coming from *food-101* dataset (cf. Section 4.2). For *Webly* data the results of a single experiment are reported with standard deviation hence this data was divided into ten equal-size parts. For *Clean* data only one value per experiment is available since this dataset was used as a whole. Each experiment was repeated 3 times with *ImageNet* initialization.

Data			Accuracy [%]			
Train	Test	Architecture	Exp.1	Exp.2	Exp.3	Mean
big webly	*Webly*	ResNet50	64.6 (+/-0.84)	64.8 (+/-0.77)	64.4 (+/-0.89)	64.6 (+/-0.83)
	Clean	ResNet50	88.5	89.4	89.4	89.1 (+/-0.53)
	Webly	VGG16	61.0 (+/-0.77)	61.0 (+/-0.54)	60.9 (+/-0.56)	61.0 (+/-0.62)
	Clean	VGG16	86.5	87.0	87.1	86.9 (+/-0.25)
small webly	*Webly*	ResNet50	62.1 (+/-0.77)	62.0 (+/-0.74)	61.9 (+/-0.72)	62.0 (+/-0.74)
	Clean	ResNet50	87.9	88.0	87.4	87.8 (+/-0.32)
clean	*Clean*	VGG16	79.0	79.7	77.8	78.9 (+/-0.79)
	Clean	ResNet50	83.5	83.6	82.9	83.3 (+/-0.28)

Table 2. *InstaCities1M*. Accuracy results for *Webly* test set and two variants of *Clean* test sets (*Random* and *Selected*) described in Section 3.7.1. *big webly* and *small webly* refer to utilization of the entire training set and half of this set, resp. (cf. Section 3.7.2). For *Webly* and *Clean Random* data the results are reported with standard deviation since these datasets were created by means of division of a larger data set into disjoint parts (*Webly*) or by sampling from a larger data set (*Clean Random*). For *Clean Selected* only one value per experiment is available since this data set is composed of 300 *easy to predict* images from each class from the *Clean* test data. Each experiment was repeated 3 times with *ImageNet* initialization.

Data			Accuracy [%]			
Train	Test	Architecture	Exp.1	Exp.2	Exp.3	Mean
big webly	*Webly*	ResNet50	32.3 (+/-0.90)	31.8 (+/-0.86)	32.0 (+/-0.81)	32.0 (+/-0.86)
	Clean Random	ResNet50	46.9 (+/-2.37)	45.1 (+/-2.37)	46.3 (+/-2.44)	46.1 (+/-2.39)
	Clean Selected	ResNet50	83.5	80.7	83.4	82.5 (+/-1.62)
	Webly	VGG16	30.3 (+/-0.75)	30.0 (+/-0.67)	30.1 (+/-0.84)	30.1 (+/-0.76)
	Clean Random	VGG16	41.9 (+/-2.43)	41.3 (+/-2.49)	42.7 (+/-2.45)	42.0 (+/-2.46)
	Clean Selected	VGG16	71.0	68.9	69.8	69.9 (+/-0.89)
small webly	*Webly*	ResNet50	29.1 (+/-0.82)	28.8 (+/-0.56)	28.9 (+/-0.78)	28.9 (+/-0.72)
	Clean Random	ResNet50	40.3 (+/-2.34)	40.4 (+/-2.34)	40.5 (+/-2.38)	40.4 (+/-2.35)
	Clean Selected	ResNet50	75.3	76.3	74.9	75.5 (+/-0.75)



(a) *InstaFood1M* (b) *InstaCities1M* (c) *InstaPascal2M*

Figure 10. Validation curves for experiments with random and *ImageNet* initialization on *small webly* and *big webly* data sets. Each curve is an average of 3 experiments.

Table 3. *InstaPascal2M*. Accuracy results for *Webly* and *Clean* test sets described in Section 3.7.1. The results are presented in the perspective of three types of training data: *small webly* - using half of the training and validation data, *big webly* - using all this data and *clean* coming from PascalVOC2007. For *Webly* test data individual results are reported with standard deviation since this data was divided into ten equal-size parts. For *Clean* data only one value per experiment is available since this dataset was used as a whole. Each experiment was repeated 3 times with *ImageNet* initialization.

Data			Accuracy [%]			
Train	Test	Measure	Exp.1	Exp.2	Exp.3	Mean
big webly	*Webly*	*acc_filtered*	51.9 (+/-1.09)	52.1 (+/-0.95)	51.9 (+/-0.96)	51.9 (+/-1.00)
		acc_one	52.1 (+/-0.77)	52.3 (+/-0.62)	52.1 (+/-0.62)	52.2 (+/-0.67)
small webly	*Webly*	*acc_filtered*	49.3 (+/-0.91)	50.2 (+/-0.82)	49.4 (+/-0.78)	49.6 (+/-0.84)
		acc_one	49.5 (+/-0.61)	50.5 (+/-0.70)	49.6 (+/-0.51)	49.9 (+/-0.61)
big webly	*Clean*	*acc_filtered*	82.7	84.7	84.1	83.8 (+/-1.03)
		acc_one	79.9	80.7	80.9	80.5 (+/-0.54)
small webly	*Clean*	*acc_filtered*	80.4	82.1	80.0	80.5 (+/-1.11)
		acc_one	77.9	80.0	77.2	78.4 (+/-1.46)
clean	*Clean*	*acc_filtered*	58.0	63.9	59.0	60.3 (+/-3.21)
		acc_one	66.3	69.0	68.1	67.8 (+/-1.34)

accuracy although the gain is moderate (2 – 5 p.p., depending on the data set) compared to the increase in the amount of data (which was doubled). It can also be observed in Figures 10a, 10b and 10c that learning curves of the networks initialized with *ImageNet* data end up stabilizing on a similar level to those initialized randomly with only up to 5% difference in the accuracy at the later epochs. The main difference, which is not surprising in fact, is the clearly longer 'warm-up' phase in randomly initiated experiments.

In summary, the learning curves confirm that **the same conclusions related to the efficacy of *webly* data training can be drawn irrespective of the weight initialization scheme (pre-training on *ImageNet* or random initialization).**

4.4 Error analysis

In food classification the most common errors were caused either by simultaneous appearance of two or more products in the image or by products similarity that confused the system. The most commonly co-appearing class were French Fries and the most frequent classes co-occurrence was that of Burger and French Fries.

Cities classification is clearly a harder task, mainly because, unlike food, cities have very much in common e.g. rivers, skyscrapers, buildings, roads, parks, etc. and therefore many city photos are not truly representative for the location in which they have been taken. The most frequent misclassifications were predictions of New York (true class: Chicago, London) and Miami (true class: Sydney, London, Chicago).

In the case of common objects classification a frequent mistake was prediction of various classes instead of a person. Additionally, quite common was confusing the classes that appear in similar surroundings, specifically mistaking a dog with a cat, a dog with a sheep or a chair with a sofa.

4.5 Potential limitations of the method applicability

The underlying assumption of proposed approach is availability of massive image-based training data in a given domain of interest. This is probably the main source of potential applicability limitations, in particular in the expert areas or less popular domains.

The other limitation, which is also domain-dependent, is frequent co-occurrence of two or more classes. Such a situation may affect final model accuracy. In our experiments with *InstaFood1M* significant part of errors were caused by simultaneous appearance of two or more products in the image. Figure 11 presents example images containing both Burger and French Fries classes, accompanied by Locally Interpretable Model-agnostic Explanations (LIME) [38] analysis (which points areas supporting prediction of a particular class in a given image, as well as those which undermine this prediction). All four images were incorrectly classified and from the LIME analysis it stems that there are significant clusters of pixels supporting each of the two classes.

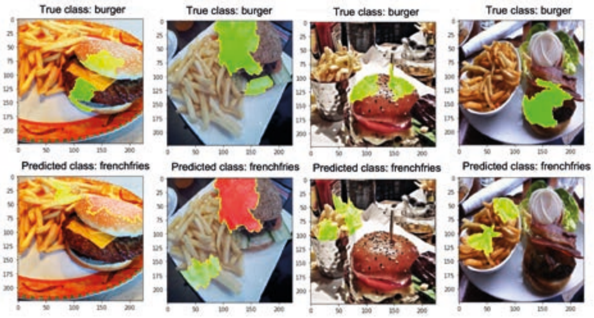

True class: burger
True class: burger
True class: burger
True class: burger
Predicted class: frenchfries
Predicted class: frenchfries
Predicted class: frenchfries
Predicted class: frenchfries

Figure 11. LIME analysis [38] of examples of the most frequently co-occurring classes: Burger and French Fries. Green color represents areas which support prediction of a given class and red color represents areas that undermine this prediction. In the upper row the analysis is presented from the perspective of Burger (true class) and in the lower one form the perspective of French Fries (wrong class).

Considering potential risks associated with practical utilization of the method, a situation which should generally be treated with special care is uneven importance of particular classes. Proposed method treats all errors with the same degree of relevance which may not necessarily be the case in practice (e.g. in medicine where false positive diagnosis is usually less harmful than false negative one). A possible remedy would be redefinition of the loss function used during training so as to reflect the true relevance of particular classes or particular misclassifications. Another risk which may materialize in practice is uneven support for the classes in the training set. A situation when some classes are represented by much smaller numbers of training examples than other classes may skew the system performance towards classes with more abundant representations. Again, the advise would be to accommodate the effects of nonuniform class distribution in the loss function or to apply data augmentation process [25] to images belonging to underrepresented classes.

5 Data cleaning

In the quest for further enhancement of results we proposed two straightforward procedures for data cleaning. In both of them, the dataset (either *InstaFood1M*, *InstaCities1M* or *InstaPascal2M*) was randomly divided into N disjoint parts $(P_1,\ldots,P_N)$ of the same size. Subsequently, N separate copies of ResNet50 network, henceforth denoted by $RN_1,\ldots,RN_N$, with the same architecture and using the same learning procedure as in the main experiments, were trained - each on one of the subsets $P_1,\ldots,P_N$, resp. Afterwards, each $RN_i, i=1,\ldots,N$ made predictions on the samples belonging to $P \setminus P_i$ where $P := \cup_{k=1,\ldots,N} P_k$. This way each image was classified $N-1$ times (by $N-1$ networks, trained independently). The following two data cleaning strategies were proposed and tested:

- **Correction** - changing the label of an image when all $N-1$ networks agreed on the same class which was different from the original one.
- **Removal** - removing an image from the training set when each of $N-1$ networks output a different class.

Both strategies aimed at removing the noise from the training data, either by correcting dubious labels or by deleting images that were presumably not representative for any class.

In all data cleaning experiments $N=5$ was used, as a reasonable compromise between results credibility (the number of concurrent predictions) and relevance of the training subsets (their reasonable sizes).

Each experiment was performed according to the following scenario: (1) random division of dataset P into $P_i, i=1,\ldots,5$, (2) training $RN_i, i=1,\ldots,5$ based on P_i, (3) testing trained RN_i on $P \setminus P_i, i=1,\ldots,5$, (4) labels correction or removal (depending on the considered cleaning variant) leading to a *cleaner* dataset P', (5) training ResNet50 on P' and its testing according to the main experiment scheme.

A summary of data cleaning experiments is presented in Table 4. The level of correction is comparable among all three datasets in terms of the percentage of corrected observations. Examples of corrected samples, presented in Figures 12, 14 and 16, respectively, show that the newly assigned labels are generally well chosen.

The number of removed samples with respect to the dataset size varies much more than in the correction experiments. *InstaCities1M* has potentially the most ambiguous labels (many city images can be non-representative for a particular city) and *InstaPascal2M* classes are the most distinct from one another. Examples of removed samples (due to complete disagreement among networks) are depicted in Figures 13, 15 and 17, respectively. The vast majority of these samples are not representative for *any* of the classes in the respective datasets and are clearly good candidates for a deletion.

Table 4. Summary of changes introduced to original datasets in effect of application of data cleaning procedures. Rows labeled *Original*, *Correction* and *Removal* indicate the initial size of the training datasets, the number of samples with modified (corrected) labels and the number of removed observations, resp. Values in parentheses show the number of observations changed/removed as a percentage of the original dataset.

Dataset	*InstaFood1M*	*InstaCities1M*	*InstaPascal2M*
Original	800 000	800 000	1 600 000
Correction	42 502	44 392	98 091
	(5.3%)	(5.5%)	(6.1%)
Removal	82 096	126 560	90 430
	(10.3%)	(15.8%)	(5.7%)

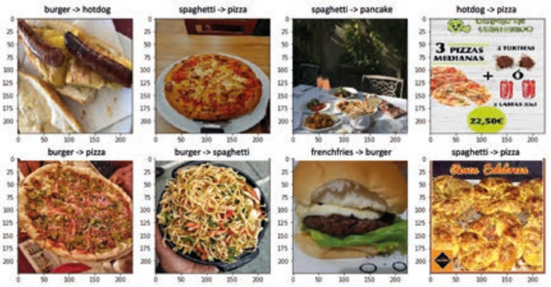


Figure 12. *InstaFood1M* examples for which all four *RN* networks agreed on the same *new* class (initial → corrected).

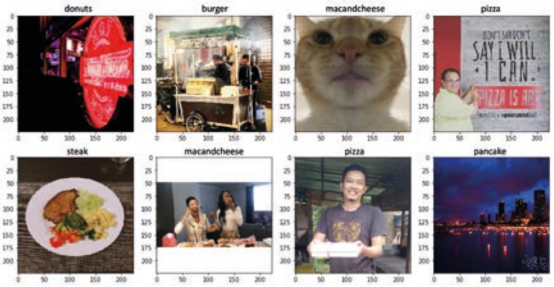


Figure 13. *InstaFood1M* examples (with initial class) for which all four *RN* networks predicted different classes.

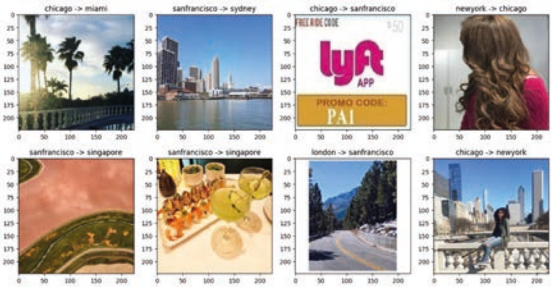


Figure 14. *InstaCities1M* examples for which all four *RN* networks agreed on the same *new* class (initial → corrected).

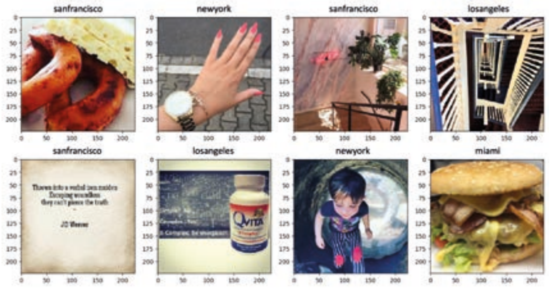


Figure 15. *InstaCities1M* examples (with initial class) for which all four *RN* networks predicted different classes.

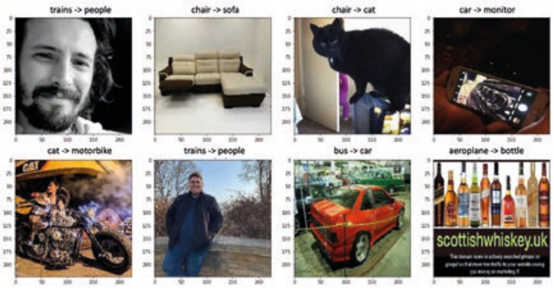


Figure 16. *InstaPascal2M* examples for which all four *RN* networks agreed on the same *new* class (initial → corrected).

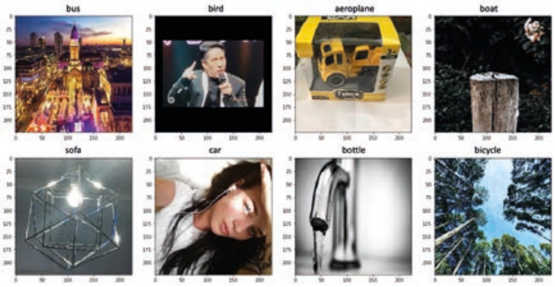


Figure 17. *InstaPascal2M* examples (with initial class) for which all four *RN* networks predicted different classes.

Quite surprisingly, the data cleaning procedures did not bring accuracy improvement over the base results (without cleaning). This outcome suggests that *noisy* images, when considered in a sufficiently large number, carry certain background information which is indeed relevant for the ultimate classification accuracy.

6 Conclusions

The paper demonstrates that it is possible to efficiently train two CNN classifiers from completely different families (VGG and ResNet) on noisy web data. For the best model, the average accuracy results on representative test data reached 89.1% on *InstaFood1M* and 82.5% on more demanding *Instacities1M* dataset. In the case of *InstaPascal2M* the results on representative test data attained (on average) 83.8% and 80.5% according to $accuracy\_filtered$ and $accuracy\_one$ measures, resp. The above scores are repeatable with very low standard deviation which supports the claim about robustness of proposed training approach.

Overall, the results confirm the possibility to use abundant weakly-labeled Internet resources of images as a source of data in the training process, with no need for manual data inspection, data cleaning or other enhancement.

The experiments showed that the resulting accuracy of webly-trained CNN classifiers is independent of the initialization method. A direct comparison of randomly initialized architectures vs. the same architectures initialized on *ImageNet* data confirmed that webly data contains all the information required to train the models effectively, although using a good starting point (initialization of weights based on *ImageNet*) speeds up the training process significantly.

An auxiliary data cleaning process did not cause accuracy improvement which suggests that class representations learnt from *webly* data are indeed meaningful and robust. In particular, excluding huge chunks of "the noisiest" data in the Removal experiments (with no consequent performance improvement) suggest that it is not a matter of the amount of noise in the *webly* training data but rather the quality of the test data that contributes mostly to the overall classification results.

Extended experiments with food-related and object-related images confirmed that training solely on class-representative data (i.e. well-framed and unambiguous images) may not be competitive to training with noisy webly data, unless the *clean* training dataset is sufficiently large. Consequently, the proposed training regime may offer a viable alternative in domains with scarce availability of expert-labeled data.

The approach presented in this paper is rudimentary, and does not require any expert-labeled data. We believe that future research on webly data utilization should follow the same path, but with a focus on a more effective usage of the information represented in webly data. In this context, interesting areas of research are methods that can either learn more effectively under the presence of noise, methods capable of selecting most useful images from webly data, and self-supervised methods (e.g. CLIP model [35]).

References

[1] J. A. Aghamaleki and S. M. Baharlou. Transfer learning approach for classification and noise reduction on noisy web data. Expert Syst. Appl., 105:221–232, 2018.

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013, pages 819–826. IEEE Computer Society, 2013.

[3] S. Bai and S. An. A survey on automatic image caption generation. Neurocomputing, 311:291–304, 2018.

[4] A. Bergamo and L. Torresani. Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada, pages 181–189. Curran Associates, Inc., 2010.

[5] J. Böhlke, D. Korsch, P. Bodesheim, and J. Denzler. Lightweight filtering of noisy web data: Augmenting fine-grained datasets with selected internet images. In G. M. Farinella, P. Radeva, J. Braz,

and K. Bouatouch, editors, Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2021, Volume 5: VISAPP, Online Streaming, February 8-10, 2021, pages 466–477. SCITEPRESS, 2021.

[6] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101 – mining discriminative components with random forests. In European Conference on Computer Vision, 2014.

[7] X. Chen and A. Gupta. Webly supervised learning of convolutional networks. In 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, pages 1431–1439. IEEE Computer Society, 2015.

[8] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, pages 2758–2766. IEEE Computer Society, 2015.

[9] T. Durand, N. Thome, and M. Cord. WELDON: weakly supervised learning of deep convolutional neural networks. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 4743–4752. IEEE Computer Society, 2016.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International Journal of Computer Vision, 88(2):303–338, June 2010.

[11] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pages 2121–2129, 2013.

[12] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014, pages 580–587. IEEE Computer Society, 2014.

[13] R. Gomez. Instacities1m, https://gombru.github.io /2018/08/01/InstaCities1M/, 2018.

[14] R. Gomez, L. Gómez, J. Gibert, and D. Karatzas. Learning to learn from web data through deep semantic embeddings. In L. Leal-Taixé and S. Roth, editors, Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part VI, volume 11134 of Lecture Notes in Computer Science, pages 514–529. Springer, 2018.

[15] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 2315–2324. IEEE Computer Society, 2016.

[16] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 2980–2988. IEEE Computer Society, 2017.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 770–778. IEEE Computer Society, 2016.

[18] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pages 10477–10486, 2018.

[19] H. Izadinia, B. C. Russell, A. Farhadi, M. D. Hoffman, and A. Hertzmann. Deep classifiers from image tags in the wild. In G. Friedland, C. Ngo, and D. A. Shamma, editors, Proceedings of the 2015 Workshop on Community-Organized Multimodal Mining: Opportunities for Novel Solutions, MMCommons 2015, Brisbane, Australia, October 30, 2015, pages 13–18. ACM, 2015.

[20] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. Int. J. Comput. Vis., 116(1):1–20, 2016.

[21] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache. Learning visual features from large weakly supervised data. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam,

The Netherlands, October 11-14, 2016, Proceedings, Part VII, volume 9911 of Lecture Notes in Computer Science, pages 67–84. Springer, 2016.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[23] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III, volume 9907 of Lecture Notes in Computer Science, pages 301–320. Springer, 2016.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States, pages 1106–1114, 2012.

[25] D. Lewy and J. Mańdziuk. An overview of mixing augmentation methods and augmentation strategies. Artificial Intelligence Review, 2022.

[26] D. Lewy and J. Mańdziuk. Instafood1m, https://szefkuchni.github.io/InstaFood1M/, 2019.

[27] D. Lewy and J. Mańdziuk. Instapascal2m, https://szefkuchni.github.io/InstaPascal2M/, 2019.

[28] J. Li, Y. Song, J. Zhu, L. Cheng, Y. Su, L. Ye, P. Yuan, and S. Han. Learning from large-scale noisy web data with ubiquitous reweighting for image classification. IEEE Trans. Pattern Anal. Mach. Intell., 43(5):1808–1814, 2021.

[29] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V, volume 8693 of Lecture Notes in Computer Science, pages 740–755. Springer, 2014.

[30] D. Mahajan, R. B. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. Exploring the limits of weakly supervised pretraining. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II, volume 11206 of Lecture Notes in Computer Science, pages 185–201. Springer, 2018.

[31] L. Niu, W. Li, D. Xu, and J. Cai. Visual recognition by learning from web data via weakly supervised domain generalization. IEEE Trans. Neural Networks Learn. Syst., 28(9):1985–1999, 2017.

[32] L. Niu, Q. Tang, A. Veeraraghavan, and A. Sabharwal. Learning from noisy web data with category-level supervision. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 7689–7698. Computer Vision Foundation / IEEE Computer Society, 2018.

[33] L. Niu, A. Veeraraghavan, and A. Sabharwal. Webly supervised learning meets zero-shot learning: A hybrid approach for fine-grained classification. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 7171–7180. Computer Vision Foundation / IEEE Computer Society, 2018.

[34] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: A loss correction approach. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 2233–2241. IEEE Computer Society, 2017.

[35] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In M. Meila and T. Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, volume 139 of Proceedings of Machine Learning Research, pages 8748–8763. PMLR, 2021.

[36] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 779–788. IEEE Computer Society, 2016.

[37] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In C. Cortes, N. D.

Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 91–99, 2015.

[38] M. T. Ribeiro, S. Singh, and C. Guestrin. ”why should I trust you?”: Explaining the predictions of any classifier. In B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, editors, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, pages 1135–1144. ACM, 2016.

[39] S. Ruder. An overview of gradient descent optimization algorithms. CoRR, abs/1609.04747, 2016.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. Int. J. Comput. Vis., 115(3):211–252, 2015.

[41] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 39(4):640–651, 2017.

[42] A. Shrivastava, A. Gupta, and R. B. Girshick. Training region-based object detectors with online hard example mining. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 761–769. IEEE Computer Society, 2016.

[43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Y. Bengio and Y. LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[44] E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in NLP. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 3645–3650. Association for Computational Linguistics, 2019.

[45] H. Su, S. Gong, and X. Zhu. Weblogo-2m: Scalable logo detection by deep learning from the web. In 2017 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2017, Venice, Italy, October 22-29, 2017, pages 270–279. IEEE Computer Society, 2017.

[46] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 843–852. IEEE Computer Society, 2017.

[47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, pages 1–9. IEEE Computer Society, 2015.

[48] A. Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5596–5605, 2017.

[49] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. J. Belongie. Learning from noisy large-scale datasets with minimal supervision. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 6575–6583. IEEE Computer Society, 2017.

[50] Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning - the good, the bad and the ugly. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 3077–3086. IEEE Computer Society, 2017.

[51] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. Learning from massive noisy labeled data for image classification. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, pages 2691–2699. IEEE Computer Society, 2015.

[52] J. Yang, X. Sun, Y. Lai, L. Zheng, and M. Cheng. Recognition from web data: A progressive filtering approach. IEEE Trans. Image Process., 27(11):5303–5315, 2018.

[53] I. Yildirim, T. Kulkarni, W. A. Freiwald, and J. B. Tenenbaum. Efficient and robust analysis-by-synthesis in vision: A computational framework, behavioral tests, and modeling neuronal representations. In Annual Conference of the Cognitive Science Society, 2015.

[54] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 3320–3328, 2014.

[55] C. Zhang, Y. Yao, X. Xu, J. Shao, J. Song, Z. Li, and Z. Tang. Extracting useful knowledge from noisy web images via data purification for fine-grained recognition. In H. T. Shen, Y. Zhuang, J. R. Smith, Y. Yang, P. Cesar, F. Metze, and B. Prabhakaran, editors, MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021, pages 4063–4072. ACM, 2021.

[56] B. Zhuang, L. Liu, Y. Li, C. Shen, and I. D. Reid. Attend in groups: A weakly-supervised deep learning framework for learning from web data. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 2915–2924. IEEE Computer Society, 2017.

Dominik Lewy received M.Sc. degree in Quantitative Methods in Economics and Information Science from Warsaw School of Economics in 2017, currently a Ph.D. degree candidate in Warsaw University of Technology. His main research interest is digital image processing in context of facilitating adoption of deep learning algorithms in business context where training data is scarce or non-existing. Commercially he is leading an Artificial Intelligence team focused on all Deep Learning related topics in Lingaro Group.
https://orcid.org/0000-0003-2107-4909

Prof. Jacek Mańdziuk, Ph.D., D.Sc., received M.Sc. (Honors) and Ph.D. in Applied Mathematics from the Warsaw University of Technology (WUT), Poland in 1989 and 1993, resp., and D.Sc. degree in Computer Science from the Polish Academy of Sciences in 2000. In 2011 he was awarded the title of Professor Titular. He is a full professor at the Faculty of Mathematics and Information Science, WUT, Head of Division of Artificial Intelligence and Computational Methods, and Head of Doctoral Program in Computer Science at this faculty.

He is the author of 3 books and 180+ research papers. He was a Senior Fulbright Scholar at UC Berkeley and ICSI Berkeley, USA, and a recipient of the Robert Schuman Foundation Fellowship at CNRS, Besancon, France. Recently, he was a visiting professor at Nanyang Technological University (Singapore), University of New South Wales (Australia), Yonsei University (South Korea) and University of Alberta (Canada). He serves/served as an Associate Editor for the IEEE Transactions on Neural Networks and Learning Systems, the IEEE Transactions on Computational Intelligence and AI in games, and the ACM Computing Surveys.

His research interests include application of Computational Intelligence and Artificial Intelligence methods to games, dynamic and bilevel optimization problems, and human-machine cooperation in problem solving. He is also interested in the development of general-purpose human-like learning and problem-solving methods. For more information please visit
http://www.mini.pw.edu.pl/~mandziuk
https://orcid.org/0000-0003-0947-028X