This is the peer reviewd version of the followng article:

W2WNet: A two-module probabilistic Convolutional Neural Network with embedded data cleansing functionality / Ponzio, F., Macii, E., Ficarra, E., Di Cataldo, S.. - In: EXPERT SYSTEMS WITH APPLICATIONS. - ISSN 0957-4174. - 214:(2023), pp. 1-10. [10.1016/j.eswa.2022.119121]

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

13/11/2026 17:13

(Article begins on next page)

arXiv:2103.13107v1 [cs.CV] 24 Mar 2021

W2WNET: A TWO-MODULE PROBABILISTIC CONVOLUTIONAL NEURAL NETWORK WITH EMBEDDED DATA CLEANSING FUNCTIONALITY

Francesco Ponzio, Enrico Macii, Elisa Ficarra, Santa Di Cataldo
Politecnico di Torino, 10129 Torino, Italy
{name.surname}@polito.it

March 25, 2021

ABSTRACT

Convolutional Neural Networks (CNNs) are supposed to be fed with only high-quality annotated datasets. Nonetheless, in many real-world scenarios, such high quality is very hard to obtain, and datasets may be affected by any sort of image degradation and mislabelling issues. This negatively impacts the performance of standard CNNs, both during the training and the inference phase. To address this issue we propose Wise2WipedNet (W2WNet), a new two-module Convolutional Neural Network, where a *Wise* module exploits Bayesian inference to identify and discard spurious images during the training, and a *Wiped* module takes care of the final classification, while broadcasting information on the prediction confidence at inference time. The goodness of our solution is demonstrated on a number of public benchmarks addressing different image classification tasks, as well as on a real-world case study on histological image analysis. Overall, our experiments demonstrate that W2WNet is able to identify image degradation and mislabelling issues both at training and at inference time, with positive impact on the final classification accuracy.

***K*eywords** Image Classification · Deep Learning · Convolutional Neural Networks · Bayesian Convolutional Neural Networks · Data Cleansing

1 Introduction

Since the milestone study by Alex Krizhevsky and colleagues in 2012 [1], Deep Learning (DL), with particular emphasis on Convolutional Neural Networks (CNNs), is the state-of-the-art method for image classification in many different applications. Besides classification performance, the reason for the success of CNNs is twofold: i) the recent boost of graphical processing units (GPUs) and parallel processing, that allows to train very large models; ii) the ever-growing availability of massive annotated task-specific datasets. Nonetheless, in many realistic applications many concerns may be raised about the reliability of such datasets both in terms of image and labelling quality, and consequently on the robustness of the CNN models trained and tested on them.

As regards to image quality, standard CNNs are supposed to be fed with high quality samples. Nevertheless, in practical scenarios different kinds of image degradation can heavily affect the performance of a CNN both in the training and in the inference phase. Problems concerning image acquisition devices, poor image sensor, lighting conditions, focus, stabilization, exposure time or partial occlusion of the cameras may lead to produce low quality samples, which have been demonstrated to be one of the chief reasons for troublesome learning procedures of CNN models in many applications [2, 3, 4]. On the other hand, even though the CNN had been proficiently trained and validated on high quality data, noisy inputs can heavily affect the inference phase, and cause classification errors at run-time. A typical example are self-driving cars, where a partial occlusion of the image acquisition device may lead to misinterpret a road sign, with catastrophic consequences. In such settings, the well-known limitations of standard CNNs to broadcast information about how much the given input resembles the ones the model was trained on - and hence, whether the associated prediction should (or should not) be trusted - is also playing a major role.

Besides image quality, also collecting and associating error-free labels to a massive number of representative images to adequately train CNNs may be extremely problematic in a number of real-world applications. If we take as an example the medical domain, where available data is typically small to begin with, image annotation is always a cumbersome and time-consuming task, that is extremely error-prone. In a number of applications, inter-observer variability is even so high as to necessitate consensus strategies to aggregate annotations from several medical experts [5], which is anyway prone to mislabelling. Conversely, in a number of non-medical real-world scenarios the collection of massive labelled image datasets is relatively easy and straightforward: for example, using semi-automatic tools based on web search engines and keywords [6]. Nonetheless, even in this case concerns may be raised on the reliability of the image labels. Take as an example the JFT dataset from Google, including $300M+$ images labeled by an algorithm that uses complex mixture of raw web signals, connections between web-pages and user feedback [7, 8]: JFT annotations have been found to be 20% wrong, even after some cleansing procedures [9].

In the rest of this paper, we will refer to image degradation and to mislabelling errors respectively by the name of *measurement* and *labelling noise*.

Even though recent studies have proposed many techniques to compensate the learning degradation due to *measurement noise* [4, 2, 3] or *labelling noise* [5, 10, 9] specifically, very few researchers have developed solutions to mitigate the impact of generic noise, where the two effects may even coexist. Furthermore, there is still very little scientific understanding of how a CNN may behave in presence of noisy inputs at inference phase, i.e. when the final model is applied to a given application, and how to make a CNN model robust to unpredictable noise effects that may make the inputs considerably different to what the model was specifically trained on.

In our study, we want to focus the attention on data-perturbation irrespective of whether it is a *measurement* or a *labelling noise*, and we will refer to *spurious* (vs. *meaningful*) samples to indicate images affected by any of the two types of noise.

We therefore propose *Wise2WipedNet* (*W2WNet*), a CNN-based architecture able to i) model the distribution of spurious samples in a generic dataset, which may be corrupted by both *labelling* and *measurement* noise; ii) clearly identify the spurious samples within the training, by virtue of an adaptive pruning criterion that is fully embedded into the learning algorithm, and focus the training on the only meaningful ones; and iii) at inference time, classify never seen images into the learned categories plus one, clearly identifying noisy inputs by means of a statistically sound measure of prediction confidence (see figure 2).

Hence, our solution exploits the concept of prediction confidence in two ways: (i) during the training phase, to establish a separability criterion between the good quality (a.k.a. meaningful) and the spurious samples, that is embedded into the learning algorithm to make the network able to focus on the only meaningful ones; and (ii) during the inference phase, to improve the robustness of the model to ambiguous inputs.

To assess the goodness of our approach in different types of settings, we evaluate *W2WNet* on several state-of-the-art public benchmarks, addressing different image classification tasks and types of noise. In addition to that, we also provide a real-world case study from the medical imaging domain.

The rest of the manuscript is structured as follows. In Section 2 we provide the background and state of the art of our work, and highlight our main contributions. In Section 3 we describe our proposed methodology and implementation details. In Section 4 we provide and discuss experimental results, respectively on the public benchmarks and on the real-world case study. Finally, Section 5 provides our final considerations and concludes the paper.

2 Background

As discussed in Section 1, in many real-world cases it is not so obvious to have high quality images to train a CNN with. Most likely, the network will face many issues arising from artifacts during image acquisition, transmission, or storage. This typically affects the training procedure, resulting in a degradation of the model performance [4, 2]. Thus, a considerable amount of literature has been published on learning CNNs with low quality images and noisy datasets. In surveillance applications, for instance, face recognition from low quality images is a key aspect, and many studies address learning low-quality faces [11, 12]. In [13] the authors show that CNNs behave very differently than human vision system (HVS) in handling minimal recognizable configurations (MIRCs), that is the smallest crop of an input image for which a human observer is able to provide a categorization. More specifically, standard CNNs are generally worse than humans at handling MIRCs, which are typically very small, and hence blurry and low resolved. In [4], the authors present the first large scale evaluation of deep networks on natural images affected by different types and different levels of image quality degradation. They show that the existing models are especially vulnerable to blur and noise. Finally, in [2], authors show the effects of degradation on different CNN models, proposing a network setup able to reduce the impact of specific type of perturbations.

As already discussed, besides *measurement* noise, also manual mislabelling or faulty automatic annotations may lead to unwieldy learning and lower classification performance [5, 6]. Previous studies specifically addressing *labelling* noise can be categorized into three main groups:

(i) Methods that focus on model selection or design. These methods aim at selecting the model, loss function and training procedures that are most robust to mislabelling [5]. Literature shows that most supervised loss functions are not fully robust to faulty labels [14], unless they are handled by overfitting avoidance [15, 6].

(ii) Data cleansing methods. The rationale is in this case to remove samples with incorrect labels. In this sense, voting among an ensemble of classifiers has been proven effective [5]. Other strategies include identifying mislabeled instances based on their impact on the training process. For example, [16] prune and re-label training instances by setting a threshold on the classification uncertainty, based on Monte-Carlo (MC) dropout. The challenge of this group of methods is to distinguish the informative samples from the harmful mislabeled ones [6]. In this sense, cleansing methods built on top of an uncertainty measure are known to be highly dependent on the given application (i.e. type and level of noise) and even on the architecture of the classifier [16, 5]. For instance, [16] set a fixed threshold on the uncertainty distribution retrieved from training samples, without modeling the distributions of the uncertainties of the noisy and clean images. Hence, the optimal threshold needs to be tailored to the given application, which may limit the usability in real-world scenarios.

(iii) Methods that propose classifier training and labelling noise modeling into a unified framework. This category somehow integrates the two aforementioned families. For instance, probabilistic models have been exploited to model the labelling noise and thereby improve classifiers [16]. Other methodologies aim at identifying and penalizing samples with incorrect labels during the training procedure [5].

While there is a large body of literature coping with either *measurement* or *labelling* noise individually, very little efforts have been directed so far to handling both the problems at one time. Nonetheless, this is a non-trivial issue in most real-world applications, where a-priori knowledge about the type of noise affecting the data may not be available. Moreover, while *labelling* noise affects the only training phase, as the supervised learning requires an appropriate labelling of the training samples, *measurement* noise may affect CNNs even at the inference phase. As already mentioned in Section 1, this may leads standard CNNs to catastrophic failure in several real-world applications. Starting from these considerations, we propose a methodology (a.k.a. *W2WNet*) able on one hand to deal with both *measurement* and *labelling* noise, and, on the other hand, to provide a statistically sound measure of prediction confidence at inference phase. Our methodology follows in the footsteps of the earlier work by [16], where the authors exploit uncertainty measures retrieved by MC dropout to identify and remove mislabelled samples. Nevertheless, we are substantially different from [16] in the following: (i) we tackle both *measurement* and *labelling* noise in parallel; (ii) we propose an end-to-end framework, embedded into a single CNN model; (iii) we provide a pruning strategy for the spurious samples which is totally automatized and adaptive to the given application; (iv) we exploit prediction uncertainty in two different ways. First, to model, recognize and remove the spurious samples from the training strategy. Second, to broadcast information on the prediction confidence, which is exploited to make CNNs robust to noisy inputs at inference time.

3 Methods

As represented in Figure 1, our architecture includes two main modules:

(i) the *Wise*, that is in charge of a two-fold aim: on one hand, to provide a reliable measure of predictive uncertainty associated to samples (Figure 1(a)); on the other hand, to model the distribution of the spurious samples for the purpose of removing them from the training dataset (Figure 1(c)).

(ii) the *Wiped*, that is the expert system trained on the cleaned dataset and designated to the actual classification phase (Figure 1(b)).

3.1 The Wise: uncertainty estimation

As already mentioned, the *Wise* must be a noise-aware model, able to associate to each prediction a corresponding uncertainty measure. Last trends in deep learning show a growing body of literature around the theme of uncertainty estimation for predictive classification models [17, 18, 16]. With special regards to CNNs, the canonical *softmax* score is erroneously regarded as a measure of prediction confidence, that is: the lower the output of the softmax, the higher the uncertainty on the corresponding prediction. Nonetheless, it has been shown that this is not true, as the softmax merely acts as a normalization [18, 19]. As a consequence, a traditional CNN might provide confident (wrong) predictions even on samples that are completely unrelated to what it was specifically trained for.

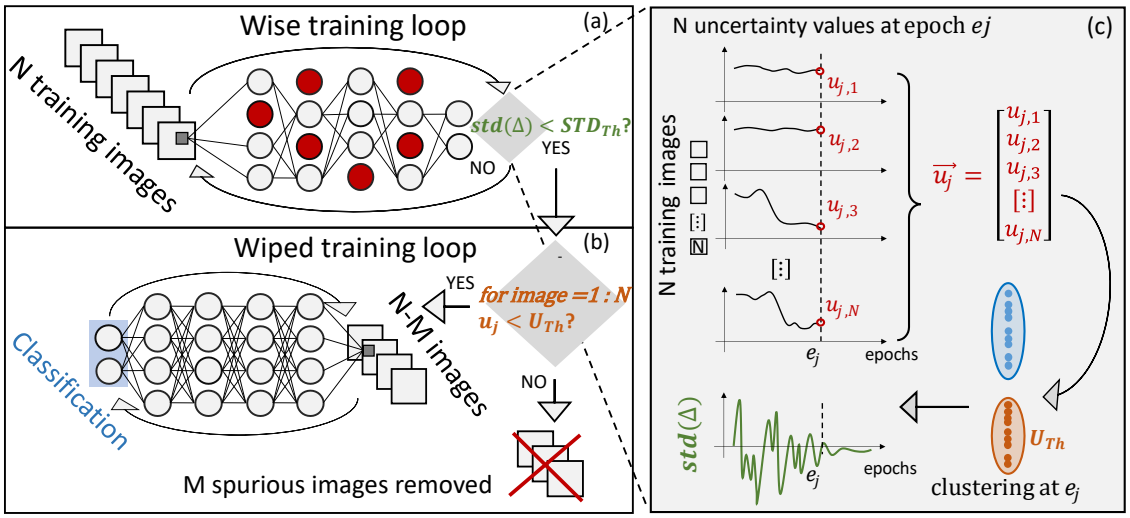


Figure 1: Overview of the training phase of the proposed architecture.

The most consolidated way to incorporate uncertainty estimation into a CNN leverages on Bayesian formalism [18, 20]. In a Bayesian perspective, individual parameters values (i.e. the weights of the network) are replaced with prior distributions. Hence, the learning strategy is conceived as a probabilistic optimization problem, where the posterior distribution over the parameters is computed, given the training data. As a consequence, the output of the model will also be a posterior predictive distribution of values, from which a statistic can be derived to serve as uncertainty measure.

Formally, the weights ω of a CNN are handled as random variables, and assuming the CNN to be exhaustively described by its weights ω, we can write the predictive distribution for a new input x^* as [18, 20]:

$$p(y^*|x^*, X, Y) = \int_{\Omega} p(y^*|x^*, \omega) p(\omega|X, Y) d\omega, \tag{1}$$

Since the term $p(\omega|X, Y)$, integrated upon the whole parameters space Ω, makes the predictive posterior of a CNN analytically and numerically intractable [17, 20], a variety of approximations have been proposed, including Laplace approximation [21], Markov chain Monte Carlo (MCMC) methods [22] and variational Bayesian methods [23, 24]. Nevertheless, the reliability of the uncertainty measure derived from these approximation strategies strictly depends on two different factors: (i) the approximation quality constrained by computational requirements; (ii) the choice of the Bayesian prior, which can ultimately lead to biased predictive uncertainties [17]. In practical terms, Bayesian CNNs (BCNNs) are cumbersome to implement and hard to train, as they require a specific training pipeline handling a very high number of possible hyper-parameters, as well as the high computational cost of the approximation technique [17]. An interesting insight by Gal and Ghahramani [18] suggested using Monte Carlo dropout (MC dropout) to estimate predictive uncertainty, which is based on using Dropout [25] at inference time. Since many different neurons are randomly dropped across different model calls, MC dropout method implements a Bayesian sampling from a variational distribution of models. In other words, MC dropout can be seen as an ensemble methodology, where the predictions are averaged over an ensemble of CNNs sharing the same parameters. In such setting, estimating the model uncertainty for a given sample is as simple as keeping the dropout mechanism switched on at inference time, and performing multiple predictions for the same input [17]. By using MC dropout, we can then rewrite equation (1) with the following approximation:

$$p(y^*|x^*, X, Y) \approx \int_{\Omega} p(y^*|x^*, \omega) q(\omega) d\omega \approx \frac{1}{T} \sum_{t=1}^{T} p(y^*|x^*, \hat{\omega}), \tag{2}$$

Thanks to variational inference [18, 26], we can approximate the posterior distribution $p(\omega|X, Y)$ in (1) with a variational one $q(\omega)$. Hence, by means of MC dropout, we assume $q(\omega) \sim \hat{\omega}$, where $\hat{\omega}$ is an estimation resulting from a variational dropout call.

Starting from the above-mentioned considerations, our *Wise* module (see figure 1(a)) was implemented as a BCNN, leveraging MC dropout. As anticipated in the previous section, the initial task of the *Wise* is to provide an uncertainty measure for the inputs samples, on the top of which the model can distinguish the spurious samples from the meaningful ones. Downstream of the uncertainty estimation, the *Wise* is able to: (i) identify and eventually remove the spurious samples, thus providing a *cleaned* dataset to train the *Wiped*; (ii) associate a confidence measure to the outcome of the *Wiped*'s classification, that can be exploited to express the reliability of the model's prediction on a given input.

To build our BCNN-based *Wise*, we put into effect equation (2) through a DenseNet121 model [27], inserting a dropout layer with 0.3 rate after each convolutional, pooling and fully connected layer. The DenseNet-based architectures connect all layers directly with each other: each layer obtains additional inputs from all preceding layers and forwards on its own feature-maps to all downstream ones [27]. By exploiting such *feature reuse* paradigm, DenseNets typically offer exceptional classification capabilities with reduced number of parameters. As it was recently observed that models with less parameters are generally more resilient to image degradations [2], we chose DenseNet121 as best trade-off between classification performances and model compactness. Nonetheless, our *Wise* module can be easily converted into any other state-of-the-art CNN architecture, by simply exploiting MC dropout instead of softmax.

Before being fed to our model, which is random initialized, samples are pre-processed by zero-centered normalization. The *Wise* is trained with Stochastic Gradient Descent (SGD), setting weight decay to 0.001. The number of training epochs of the *Wise* model is a key parameter, which is self-optimized as explained in the next section.

Ultimately, we need to define a statistically sound measure of uncertainty. To do so, we adopted the methodology proposed by Kwon and colleagues [20]: starting from (2), the predictive uncertainty of a BCNN may be computed as the sum of the predictive variances of each class [19]. Such predictive variance can be further decomposed into the aleatoric component, able to represent the intrinsic noise in the samples, and the epistemic component, which stems from the parameters and the architecture of the model:

$$\underbrace{\frac{1}{T}\sum_{t=1}^{T} diag(\hat{p}_t) - \hat{p}_t^{\otimes 2}}_{\text{aleatoric}} + \underbrace{\frac{1}{T}\sum_{t=1}^{T}(\hat{p}_t - \bar{p})^{\otimes 2}}_{\text{epistemic}} \tag{3}$$

Here $\bar{p} = \sum_{t=1}^{T} \hat{p}_t/T$; $\hat{p} = Softmaxf(\omega_t, x^*)$ and T is the number of forward passes for input x^*. T has been empirically set to 100 as the best trade-off between computational time and reliability, as stated in [28].

3.2 The Wise: modelling of spurious samples distribution

The aforementioned uncertainty measure provides a way to distinguish between spurious and meaningful samples. The *Wise* has a two-fold functionality. On one hand, during the training phase (see Figure 1), it should identify an epoch e_j so that the uncertainty of the spurious samples is significantly higher than the uncertainty of the meaningful samples. Hence, *Wise*'s training should proceed until (i) the separation between high uncertainty (i.e. spurious) and low uncertainty (i.e. meaningful) samples is large enough, and (ii) this separation is sufficiently stable over the training epochs. On the other hand, the *Wise* must identify an uncertainty threshold U_{Th} (see Figure 1(c)) that will be exploited at inference time, to broadcast information on the level of confidence of the final prediction (see Figure 2).

To pursue the stated goals, for a generic $j-th$ training epoch the learning proceeds as follows:

(i) The *Wise* computes for each training sample a corresponding classification uncertainty value, by means of equation (3); thus, given N training samples, we obtain a vector of N uncertainty values, referred to as $\vec{u_j}$ in Figure 1(c);

(ii) The vector $\vec{u_j}$ is given as input to a K-means clustering, with $K = 2$, where the low-uncertainty and the high-uncertainty clusters should represent the meaningful and spurious clusters, respectively. After doing so, the difference between the two clusters' sizes is computed and normalized upon the total number of training samples. Hence, after j training epochs, we obtain a signal δ made of j such values, whose evolution over time can be exploited to estimate the stability of the clustering at the given epoch. That is, the more stable δ is over the epochs, the lower the number of samples that are re-assigned to a different cluster, and hence, the more stable the clustering;

(iii) At this stage, we need a quantitative stability criterion to stop the *Wise*'s training. First, δ is low-pass filtered via a median filter with a window size of 11. Second, the standard deviation is computed over a sliding window of size 40 and a stride of 1, obtaining the signal referred to as $std(\Delta)$ at the bottom of Figure 1(c)). To decide on the stability of the clustering at epoch e_j, and hence on whether to stop the training, $std(\Delta)$ is imposed a

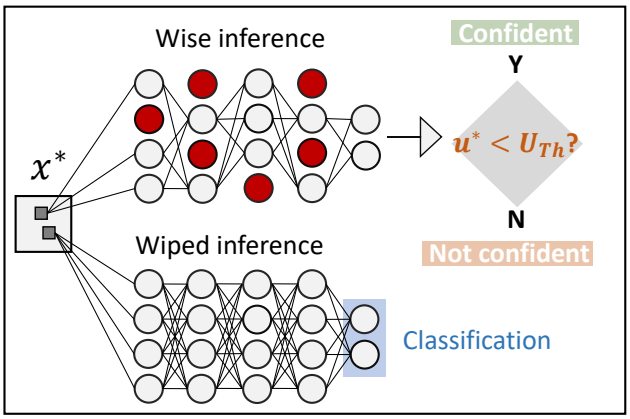


Figure 2: Overview of the inference phase of the proposed architecture.

threshold STD_{Th}, which is set to 0.01 (see Figure 1 (a)). In other words, we stop the training of the *Wise* if more than 99% of the training samples are stably assigned to the same cluster for 40 consecutive epochs. At inference time, the centroid of the spurious cluster will be exploited as an uncertainty threshold, referred to as U_{Th}, in order to identify the samples upon which the model's prediction is not sufficiently confident.

3.3 The Wise & the Wiped: classification

While providing a framework to estimate prediction uncertainty, standard BCNNs are often less accurate than their deterministic counterparts at inference time [29, 28]. To address this issue, as it can be gathered from Figure 2, in our model both the *Wise* and the *Wiped* take part in the inference phase. Given a classification task involving C classes and a generic test sample x^*, the *Wise* initially computes the corresponding uncertainty u^* through equation 3. Then, u^* is compared with the threshold U_{Th}, identifying x^* either as a *confident* or a *not-confident* prediction. Beside this first categorization, the *Wiped* will also assign a classification label in the range $[1, C]$ to x^*.

Beneath the lid, the *Wiped* module is a canonical deterministic DenseNet121 model, trained on the only meaningful samples as pre-identified by the *Wise*. The training procedure is the same that was described in Section 3.1, with the only difference that the number of epochs is fixed and equal to 100.

4 Experimental Results

In this Section we present the experimental validation of our *W2WNet*. So far, there is no agreed upon benchmark protocol to evaluate learning methods in the way they handle *measurement* and *labelling* noise. Therefore, we started from two well-known public datasets, the MNIST [30] and the CIFAR10 [31], and we artificially corrupted such datasets in a controlled way. By doing so, we tried to replicate different types of real-world noisy scenarios:

(i) *Labelling noise (labels from a different classification task).*
In text processing, handwritten character classifications are typical mainstream tasks for CNNs. The MNIST dataset, that is made of 60000 black and white images of handwritten digits (0 to 9), was corrupted by adding a controlled percentage of alien samples randomly extracted from the EMNIST dataset [32], which contains handwritten alphabetical characters. Hence, the resulting corrupted dataset, referred to as *Sp-MNIST*, contains either digits (that are still the majority of the images) and letters, all with a white foreground and black background. By doing so, we simulate a real-world situation where a pre-processing pipeline may produce spurious samples to a downstream classifier that was specifically trained on digit classification, due to text parsing errors. This scenario is similar to any other instances of data corruption, where the spurious samples share the same characteristics of the meaningful ones in terms of color range and encoding, but belong to different classifications tasks (in this case, digits and alphabets).

(ii) *Labelling and measurement noise (labels from the same classification task).*
As anticipated in Section 1, in natural image classification, datasets may be corrupted by both labelling and measurement noise. Mislabelling may sometimes occur due to errors during the automatic collection of a large amount of annotations from the Internet (for example, by extracting tags from the surrounding texts or keywords from search engines). On the other hand, measurement errors can always occur because of problems

Table 1: Validation benchmarks: number of images

Dataset	Train		Test	
	Meaningful	**Spurious**	**Meaningful**	**Spurious**
MNIST	60000	-	10000	-
$Sp-MNIST-10$	60000	6000	10000	1000
$Sp-MNIST-20$	60000	12000	10000	2000
$Sp-MNIST-30$	60000	18000	10000	3000
$CIFAR10$	50000	-	10000	-
$Sp-CIFAR10-10$	50000	5000	10000	1000
$Sp-CIFAR10-20$	50000	10000	10000	2000
$Sp-CIFAR10-30$	50000	15000	10000	3000

with acquisition and storage of the images. To simulate such scenarios, we exploited the CIFAR10 dataset, which consists of 50000 32x32 RGB images of 10 classes of natural objects. As regards to labelling, the dataset was artificially corrupted by two different types of noise patterns: symmetric and pair. In the former, original labels are randomly flipped to another label. In the latter, labels are systematically flipped to the subsequent one. Both the patterns are well know in literature, as they are experienced in several image classification tasks [16]. As regards to measurement noise, we picked a random pool of images from CIFAR10 and applied three different types of transformations: (i) blurring, via a median filter with kernel size 11; (ii) random cropping; (iii) random scaling. Even in this case, such image degradation is widely reported by literature, and known to be troublesome for CNN learning in many classification tasks [4]. As a result of our artificial corruptions, in the final dataset, referred to as *Sp-CIFAR10*, a known subset of images are either given a wrong label (which, differently from the previous case, belongs to the same classification task of the original dataset), or altered in terms of image definition, scale and dynamic range.

To push the capabilities of our methodology to its limits, for both the above-mentioned settings, we introduced increasing amount of spurious samples (respectively, 10, 20 and 30% of the size of the original dataset). A full characterization of the obtained validation datasets is reported in Table 1. In this table, each dataset is referred to as $[Sp]-name-[N]$, where the Sp prefix indicates the presence of spurious samples, $name$ is the acronym of the original dataset and N is the percentage of spurious samples with respect to the total size of the corresponding original dataset.

4.1 Data cleansing capability

As a matter of principle, our *W2WNet* should follow three fundamentals: (i) if spurious samples are present, it should remove as many as possible (i.e. high sensitivity); (ii) while removing spurious samples, it should remove as little meaningful ones as possible, as they might be essential for the training of the model (i.e. high specificity); (iii) it should be able to handle datasets that do not contain any spurious samples (that is the ideal case), and leave them untouched.

To assess all the mentioned specifications, we trained and tested our *W2WNet* both on the corrupted datasets (i.e. the ones with the Sp prefix in Table 1) as well as on the corresponding original ones, in the exact configuration of their reference papers.

In Figure 3 we show the results of our experiments. Bars show the average number of images removed per dataset, separately for the training and for the test phase. In the former case, *removed* images means that the model tagged them as spurious and hence removed them from the training set. In the latter case, the trained model tagged them as spurious at inference time, by providing a low-confidence prediction.

The first plot of Figure 3 reports the percentage of spurious samples which were correctly identified and removed from the corrupted datasets (i.e. the sensitivity of the model). The last two plots report the number of meaningful samples mistakenly tagged as spurious, respectively on the corrupted datasets and on the original ones. As mentioned earlier, the lower these numbers, the higher the specificity of the model.

As it can be gathered from the first plot, *W2WNet* was able to remove at least 30% and at best 70% of the spurious images, when considering both the training and the test sets. Apart from the training of the $Sp-CIFAR10$, where it is possible to see a decreasing trend of the bars, the performance was quite stable at increasing number of spurious samples in the datasets. The relation between the sensitivity on the training and test sets was different for the two applications: higher on the training than on the test set for the $Sp-CIFAR10$ datasets, the opposite for the $Sp-MNIST$ ones.

As it can be gathered from the second plot, *W2WNet* proved to be reasonably specific in the corrupted datasets, removing as little as 17% of meaningful samples in the worst case ($SP-CIFAR-30$) and almost 0% in the best case ($SP-MNIST$).

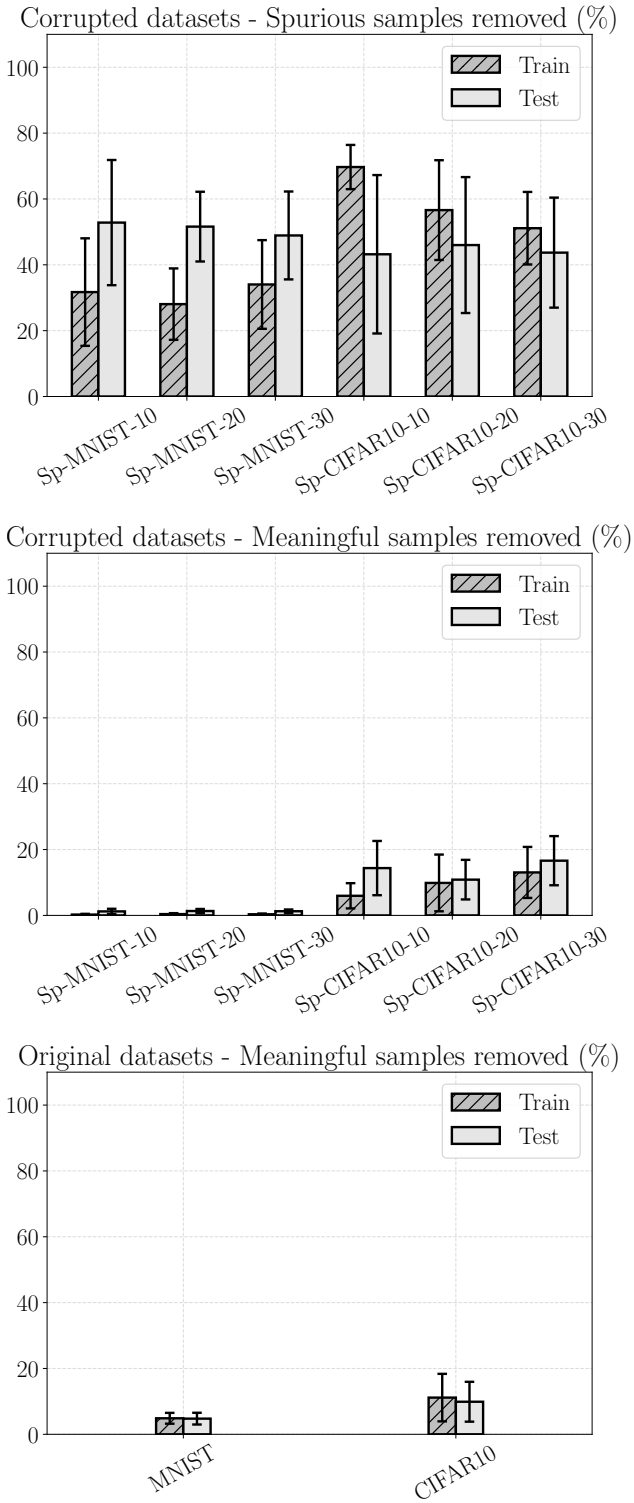


Figure 3: *W2WNet* removal rates in the validation datasets. Error bars represent standard deviation of values among different classes.

Finally, by looking at the last plot, the number of meaningful images that were on average mistaken as spurious in the original datasets were 5 and 10%, respectively in MINST and CIFAR10. A more thorough analysis revealed that in both cases these samples are very ambiguous images, that a human observer can hardly ascribe to any of the

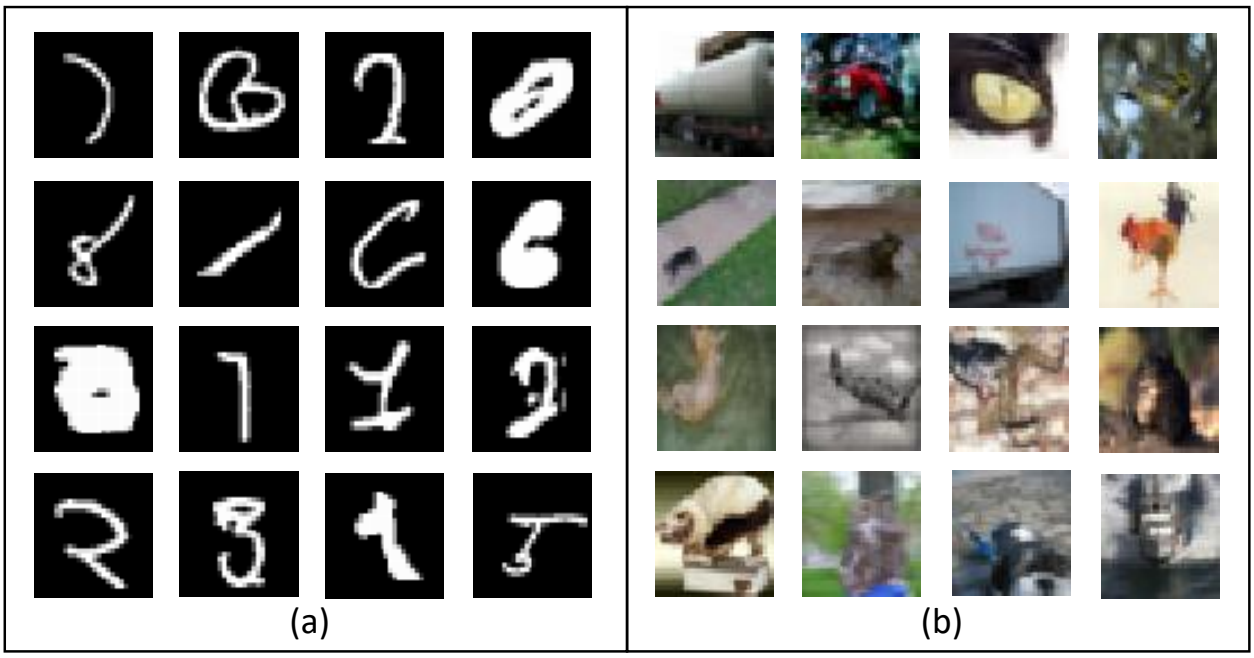


Figure 4: Examples of images tagged as *spurious*, respectively from MNIST (a) and CIFAR datasets (b).

training categories (see Figure 4). Hence, we believe that tagging such images as spurious is totally reasonable, and more importantly, it does not have a negative impact on the training, as will be showed later on. Overall, *W2WNet* is reasonably sensitive and specific in the identification of spurious samples, and the reliability of the uncertainty measure, associated with the final prediction, is proved by our results.

4.2 Classification performance

At last, to assess the effectiveness of our solution in terms of positive impact on the classification performance, we compared *W2WNet* against a canonical deterministic counterpart on all the datasets reported in Table 1. For this purpose we exploited a deterministic DenseNet121 model, as it is also the backbone of our *W2WNet* architecture, and hence it is totally equivalent to our model in terms of depth and classification potential. For the training of the deterministic CNNs, we followed the same procedure described in Section 3, with the only difference of having set the MC dropout rate to zero. The learning rate was set to 0.1 and 0.01, respectively for the datasets derived from MNIST and CIFAR10.

As already anticipated in Section 1, to the best of our knowledge, there is no published literature on deep learning methods addressing *measurement* and *labelling* noise coexisting together. Nonetheless, to better contextualize our validation, besides our approach and its deterministic counterpart, we also provide results obtained by representative algorithms facing either *measurement* or *labelling* noise. For the former category, we tested the methodology by Roy and colleagues [2], which leverages on a not trainable low-pass filter-like CNN layer to reduce the impact of image degradation on the classification performance. For the latter, we put into effect the work by Kohler et al., in the configuration made up of a single MC droput-based classifier with 25 forward passes [16]. For a fair comparison, both the methods were implemented using a DenseNet121 model as the backbone.

The results of our experiments are reported in Figure 5, where we show the mean classification accuracy obtained by the four models (our *W2WNet*, a deterministic DenseNet121, ad the two literature data cleansing approaches). As it can be observed from the plot, for all the approaches, the mean classification accuracy decreases at increasing number of spurious samples affecting the dataset (from 10 to 30 %, see also Table 1). This is absolutely consistent with previous literature [16]. When considering the corrupted datasets, our *W2WNet* outperforms the deterministic DenseNet121 of a value between 5% and 10%. In addition, *W2WNet* overcomes both the baseline literature solutions, which both behave similarly to DenseNet121. This is not surprising, as both methods are specifically tailored to address one type of noise solely. By a lesser margin, the accuracy of our *W2WNet* was the highest even in the non-corrupted datasets.

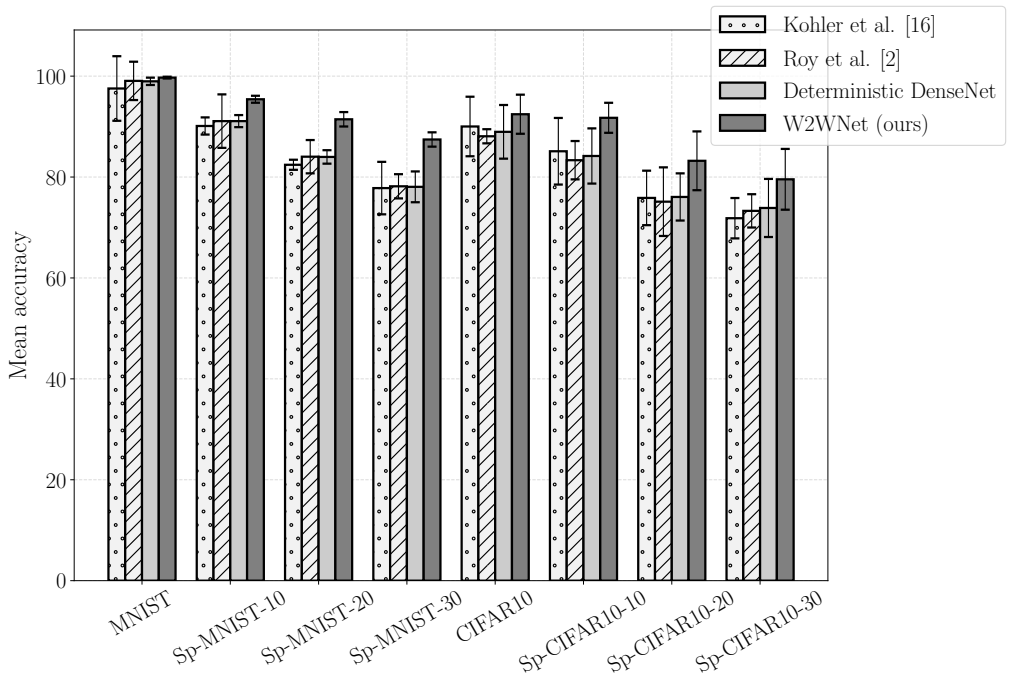


Figure 5: Mean accuracy of *W2WNet* compared with representative works from literature. Error bars represent standard deviation of values among different classes.

4.3 Real-world case study: histological images classification

Histological image analysis is the gold standard for the diagnosis and gauging of large number of cancers [33]. Typically, when there is a suspicion of cancer, the patient goes through a biopsy, where a thin layer of tissue sample is resected, fixed on a slide, and stained (for example, by Hematoxylin and Eosin). Then, the pathologist analyzes the slide on the microscope looking for malignancies, which commonly cause alterations of the normal tissue architecture. The recent diffusion of digital scanners imposed the transition from standard histological slides to very large born-digital multi-resolution images called Whole-Slide Images (WSIs, see Figure 6(a)), whose typical size may be $100,000 \times 100,000$ pixels. This is rapidly changing the workflow of clinical laboratories [34]: the traditional visual evaluation of the samples directly under the microscope is progressively shifting to Computer-Aided Diagnosis (CAD) systems, encouraging a complete automatization of downstream image analysis.

Recently, researchers have shown an increased interest in applying DL techniques (most often based on CNNs) to the automated assessment of the WSIs. Nonetheless, obtaining good quality training sets for the CNNs is an extremely cumbersome task, involving a number of steps: (i) manually dividing each WSI into regions of interest (ROIs), that should be homogeneous in terms of tissue architecture; (ii) manually labelling ROIs, based on the tissue category (e.g. cancer vs no-cancer, see Figure 6(b)); (iii) cropping ROIs into a regular grid of small tiles, that can be fed into a CNN together with their corresponding label (the same of the corresponding ROI, Figure 6(c)). Due to image artifacts, imprecision in the ROI delineation, or non-homogeneous content of the ROIs, the outcome of this procedure is typically a dataset which may contain a large amount of spurious tiles: that is, a significant number of tiles may have a content that is either too blurred (measurement noise) or unrelated to the label they were associated to (labelling noise), and then potentially harmful for the training of the CNN. For example, in Figure 6(e), a number of tiles labeled as *cancer* contain a prevalence of background glass, which is obviously not meaningful to the *cancer* category. This makes it a significant case-study for the exploitation of our *W2WNet*.

More specifically, in our experiments we refer to the same case study described in our earlier work [28], focused on Colorectal Cancer (CRC) categorization. In this case, the classes of interest are three: (i) Adenocarcinoma (AC), corresponding to recognizable CRC; (ii) Tubulovillous adenoma (AD), a precursive lesion of CRC, and (iii) Healthy tissue (H). As detailed in [28], downstream of the automated ROI cropping and labelling procedure represented in Figure 6, a total number of 19644 non-overlapping annotated tiles were obtained from 27 different WSIs. After ad-hoc re-examination of the tiles by a pathologist, 6144 of them were tagged as spurious, as the prevailing content of such slides (either blood vessels, adipose cells, background glass or stroma, see figure 6(e)), was not deemed meaningful to any of the three classes of interest.

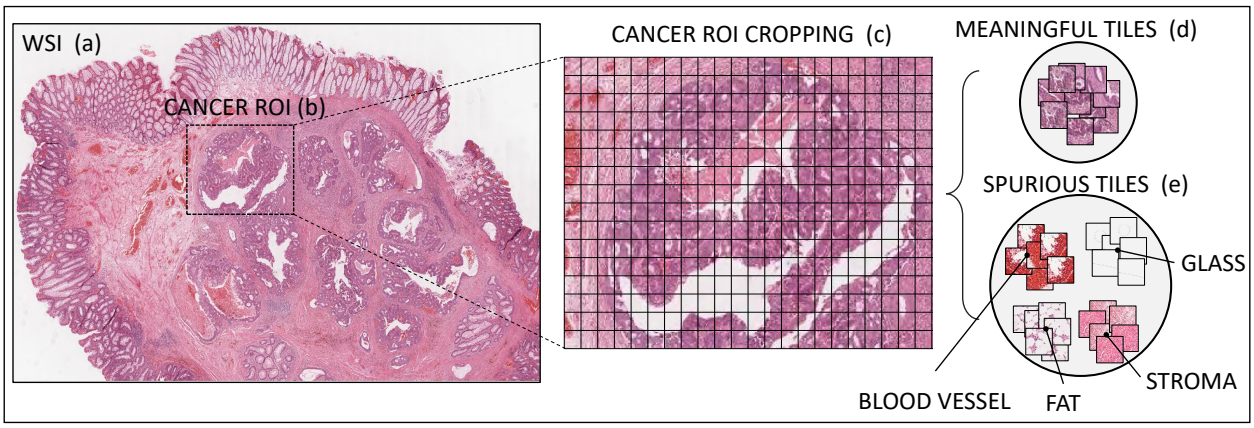


Figure 6: Generation of a digital patholohy dataset to train CNNs: typical automated procedure. (a) Whole Slide Image (WSI). (b) Identification and labelling of homogeneous Regions of Interest (ROIs). (c) Cropping ROIs into small tiles, which are all given the same label of the originating ROI. (d) Meaningful tiles (e) Spurious tiles (that is, tiles whose content is not fully representative of the given label).

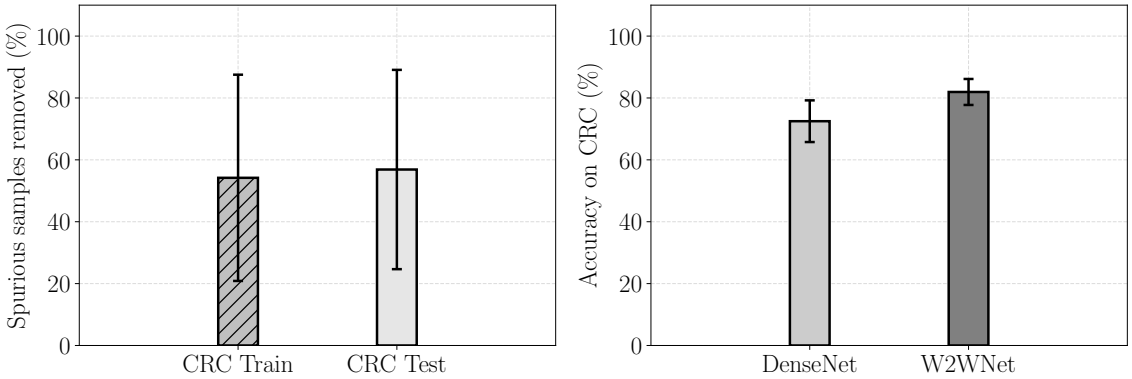


Figure 7: Removal rate of spurious samples with *W2WNet* (left) and classification accuracy comparison of *W2WNet* and DenseNet (right) on the CRC dataset. Error bars represent standard deviation of values among different classes.

For training and testing purposes, the initial cohort of 27 WSIs was randomly split into two disjoint subsets (18 for training and 9 for testing), roughly balanced with respect to the classes, and then fed into our *W2WNet* for data cleansing and classification.

The results of our experiments are shown in Figure 7. As it is visible from the plot on the left, our framework was able to identify 55% and 58% of the spurious samples from training and test set respectively. The impact on the classification is shown on the right plot, where we compare the mean classification accuracy of *W2WNet* with the one obtained by the deterministic counterpart, a state-of-the-art DenseNet121 CNN, trained from scratch with learning rate set to 0.0001 and SGD optimizer. Even in this case, the accuracy of our proposed solution was higher, by about 9% on average on the test set.

5 Conclusions

Unfortunately, *measurement* and *labelling* noise are unavoidable in many real-world applications of CNNs. On one hand, the training phase of a CNN may be affected by many types of image degradation, due to problems of acquisition, encoding or storage, and mislabelling, due to faults of the manual annotation or of the automated labelling systems. On the other hand, at inference time, a CNN that was trained on a good quality dataset may be fed with low-quality images, that are completely unrelated to the ones the model was trained on. Even in such cases, a standard CNN is neither able to provide a correct prediction, nor to communicate its impossibility to provide a reliable answer.

To address this issue, in this paper we proposed *W2WNet*, a CNN architecture exploiting Bayesian probabilistic inference to i) identify the peculiar distribution of spurious samples in a dataset, that may be affected by both measurement and

labelling noise; ii) clean the training dataset from the spurious samples and focus the learning strategy on the only meaningful ones; iii) at inference time, provide a statistically well-founded measure of prediction confidence on the new inputs, clearly identifying the ones on which the network is too uncertain.

Our experiments on MNIST and CIFAR10 datasets, artificially corrupted by a controlled number of spurious samples, has shown that *W2WNet* can cope well with measurement and labelling noise, both in terms of sensitivity and specificity in the identification of the spurious samples. As an effect of this, *W2WNet* improves on the classification accuracy of a DenseNet121 CNN, which is the deterministic counterpart of our classifier, as well as of state-of-the-art methods, which are tailored to one specific type of noise. On top of that, we found that *W2WNet* outperformed the other techniques even in the classification of non-corrupted datasets (i.e. original MNIST and CIFAR10), thanks to its capability of discarding a limited number of ambiguous images from such datasets.

Ultimately, we evaluated *W2WNet* in a real-world case study from medical image analysis, that is the classification of histological samples from WSIs. Even in this case, *W2WNet* was able to handle the presence of several spurious samples, that were generated by a typical dataset generation pipeline in digital pathology [28], and improve on the performance of the DenseNet121.

In conclusion, we believe that our findings have important implications for the proficient exploitation of DL models in many real-world settings, where the presence of image quality and labelling issues typically challenge the use of classic CNN architectures, both during the training and the inference phase.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[2] Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. Effects of degradations on deep neural network architectures. *arXiv preprint arXiv:1807.10108*, 2018.

[3] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[4] Samuel Dodge and Lina Karam. Understanding how image quality affects deep neural networks. In *2016 eighth international conference on quality of multimedia experience (QoMEX)*, pages 1–6. IEEE, 2016.

[5] Davood Karimi, Haoran Dou, Simon K Warfield, and Ali Gholipour. Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. *Medical Image Analysis*, 65:101759, 2020.

[6] Xin Liu, Shaoxin Li, Meina Kan, Shiguang Shan, and Xilin Chen. Self-error-correcting convolutional neural network for learning with noisy labels. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 111–117. IEEE, 2017.

[7] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[8] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2017.

[9] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[10] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015.

[11] Wilman WW Zou and Pong C Yuen. Very low resolution face recognition problem. *IEEE Transactions on image processing*, 21(1):327–340, 2011.

[12] Chuan-Xian Ren, Dao-Qing Dai, and Hong Yan. Coupled kernel embedding for low-resolution face image recognition. *IEEE Transactions on Image Processing*, 21(8):3770–3783, 2012.

[13] Shimon Ullman, Liav Assif, Ethan Fetaya, and Daniel Harari. Atoms of recognition in human and computer vision. *Proceedings of the National Academy of Sciences*, 113(10):2744–2749, 2016.

[14] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[15] B. Frenay and M. Verleysen. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2014.

[16] Jan M Köhler, Maximilian Autenrieth, and William H Beluch. Uncertainty based detection and relabeling of noisy image labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 33–37, 2019.

[17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413, 2017.

[18] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[19] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[20] Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik. Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis*, 142:106816, 2020.

[21] David JC MacKay. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology, 1992.

[22] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[23] Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

[24] Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning*, pages 1708–1716, 2016.

[25] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[26] Łukasz Rączkowski, Marcin Możejko, Joanna Zambonelli, and Ewa Szczurek. Ara: accurate, reliable and active histopathological image classification framework with bayesian deep learning. *Scientific reports*, 9(1):1–12, 2019.

[27] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[28] F. Ponzio, G. Deodato, E. Macii, S. Di Cataldo, and E. Ficarra. Exploiting "uncertain" deep networks for data cleaning in digital pathology. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 1139–1143, 2020.

[29] Kumar Shridhar, Felix Laumann, and Marcus Liwicki. A comprehensive guide to bayesian convolutional neural network with variational inference. *arXiv preprint arXiv:1901.02731*, 2019.

[30] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[31] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[32] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[33] Francesco Ponzio, Gianvito Urgese, Elisa Ficarra, and Santa Di Cataldo. Dealing with lack of training data for convolutional neural networks: The case of digital pathology. *Electronics*, 8(3):256, 2019.

[34] Navid Farahani, Anil V Parwani, and Liron Pantanowitz. Whole slide imaging in pathology: advantages, limitations, and emerging perspectives. *Pathol Lab Med Int*, 7(23-33):4321, 2015.