Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 217 (2023) 1323–1334

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Industry 4.0 and Smart Manufacturing

The Truth is Out There: Focusing on Smaller to Guess Bigger in Image Classification

Vagan Terziyan [a]*, Olena Kaikova [a], Diana Malyk [b], Vladyslav Branytskyi [b]

[a] *Faculty of Information Technology, University of Jyväskylä, 40014, Jyväskylä, Finland*
[b] *Department of Artificial Intelligence, Kharkiv National University of Radio Electronics, 61166, Kharkiv, Ukraine*

Abstract

In Artificial Intelligence (AI) in general and in Machine Learning (ML) in particular, which are important and integral components of modern Industry 4.0, we often deal with uncertainty, e.g., lack of complete information about the objects we are classifying, recognizing, diagnosing, etc. Traditionally, uncertainty is considered to be a problem especially in the responsible use of AI and ML tools in the smart manufacturing domain. However, in this study, we aim not to fight with but rather to benefit from the uncertainty to improve the classification performance in supervised ML. Our objective is a kind of uncertainty-driven technique to improve the performance of Convolutional Neural Networks (CNNs) for image classification. The intuition behind our suggested "decontextualize-and-extrapolate" approach is as follows: any image not necessarily contains all the needed information for perfect classification; any trained CNN will give for the entire image (with some uncertainty) the probability distribution among possible classes; the same CNN may also give similar probability distribution to the "part" of the image (i.e., with the higher uncertainty); one may discover the trend of the probability distribution change with the change of uncertainty value; a better (refined) probability distribution could be computed from these two distributions as the result of their extrapolation towards the less uncertainty. In this paper, we suggested several ways and corresponding analytics to discover reasonable part(s) of the images and to make the extrapolation to get better (refined) image classification results. We have considered image representation at the level of pixels as well as at the level of the discovered features. Our preliminary experiments show that the suggested refinement techniques (applied during the testing phase of the CNNs) can improve their classification performance.

Keywords: machine learning, deep learning, image classification, uncertainty, Convolutional Neural Network, classification refinement.

* Corresponding author.
E-mail address: vagan.terziyan@jyu.fi

1877-0509
10.1016/j.procs.2022.12.330

1. Introduction

Industry 4.0 and smart manufacturing domains are dealing nowadays with the increased uncertainty as noticed by Longo et al. [1]. Much of the uncertainty is brought there due to various human factors [2]. AI and ML methods, tools and systems are deeply embedded nowadays into Industry 4.0 [3]. Therefore, an essential share of the uncertainty is also brought by them. A good classification of ML uncertainty issues in industry applications is available in [4]. The vast majority of related studies in ML are intended to combat, tame, decrease or handle uncertainty [5]. In contrast, in this study, we are going to artificially increase uncertainty within some ML tasks and benefit from that when appropriate.

For our study we consider the supervised ML domain and particularly image classification with CNNs. Uncertainty (to a certain extent) is always presented in image classification tasks because an image may not include neither explicitly nor implicitly all the needed information for its perfect classification. However, we may use here the philosophy of the "semantic balance existence" between the internal and external knowledge axiom [6]. Among other consequences of this axiom, an interesting thing for this study could be the following one. If you have incomplete information within the boundaries of known and relevant facts needed for your objectives, which provokes uncertainty in using these facts efficiently, and if it is also impossible to get any extra information outside these boundaries, then you may still try to get useful extra information from inside the boundaries. One way of doing that would be as follows: take a subset of available information (e.g., cut some part of the image and, therefore, artificially increase its uncertainty); solve your task (e.g., classify) twice: with the entire knowledge (original image with some "reasonable" uncertainty) and with the chosen part(s) (with "larger" uncertainty); combine both outcomes in the "extrapolation" way, i.e., moving from "larger" uncertainty to the intended "perfectness" through the "reasonable" uncertainty. To make this trick work, we need certain analytics to measure uncertainty, discover suitable parts ("decontextualize") of images, combine ("extrapolate") the classification outcomes for the entire images and their parts to get the refined classification outcome with potentially better classification accuracy. Just these objectives (under the overall umbrella of our "decontextualize-and-extrapolate" approach to classification refinement) we are going to address in this paper. We also want to emphasize that "getting part(s)" of images in this study must not be confused with either semantic segmentation of images [7] or with the CNNs' pruning techniques [8].

The rest of the paper is organized as follows: Section 2 provides a motivation scenario for our study; Section 3 discusses a naïve and biased refinement option just with one part of the input image; Section 4 makes the refinement less biased by considering several parts of the image discovered by "sliding focus"; Section 5 suggests using entropy measure to discover the best parts of the image for the refinement; Section 6 considers refinement schemas, which will work not at the level of pixels but at the level of the discovered features; Section 7 summarizes our experiments, which compare classification performance of different refinement techniques; and we conclude in Section 8.

2. Looking beyond an image (the motivating scenario)

Assume that we are classifying some abstract object (e.g., an image of a dog marked as "full" in Fig. 1) on the basis of available input information about the object (e.g., a complete set of pixels representing the image) to one of two possible classes (e.g., "dog" and "cat"). Assume that the amount of information encoded in the input is ω_{full} (it could be measured by different ways, e.g., simply the number of pixels representing the complete image). The result of classification by some trained classifier (e.g., by CNN) would be a probability distribution among the considered classes. Let it be as follows (Fig. 1):

$$\{p_{dog}^{full} = 0.45; p_{cat}^{full} = 0.55\},$$

and it indicates that the classifier is not quite certain about preferring the "cat" over the "dog" label, which is actually incorrect.

Assume that we take some subset of the input information, i.e., "decontextualize" part of the input (e.g., cut part of the image and get an image marked as "part" in Fig. 1). Naturally, the amount of information within the decontextualized part will be less than in the whole input, i.e.: $\omega_{part} < \omega_{full}$.

Assume now that we apply the same trained classifier to label the part as a separate input and we get another probability distribution among the classes (see Fig. 1):

$\{p_{dog}^{part} = 0.2; p_{cat}^{part} = 0.8\}$,

which indicates that the chosen part of the input gives more evidence for the classifier preferring "cat" over "dog" and, therefore, making the misclassification error even greater.

An interesting question would be: what if we somehow get some extra information about our object, so that the resulting (the original plus the new one) set of input information "res" will be greater than "full", i.e.: $\omega_{res} > \omega_{full}$, then what would be the result of classification with such greater evidence? An important yet intuitive assumption here could be as follows: (a) when dealing with "full" and "part" (Fig. 1), we noticed that with getting more information ("part" → "full") the "dog" classification choice has the tendency of growing (0.2 → 0.45) while the probability of the "cat" option decreases (0.8 → 0.55); (b) if the tendency stays the same with getting more information, we may expect that the probability for the "cat" option will decrease from 0.55 further and probably falls to a value less than 50%, while the probability for the "dog" option will continue increasing from 0.45 further and could result to the value greater than 50%; (c) without any actual update of input information and just by studying the ("part" → "full") tendencies ("extrapolating"), we may refine and even change when appropriate the result of classification. For example, Fig. 1 shows that such an extrapolation …:

$$\{p_{dog}^{part} = 0.2; p_{cat}^{part} = 0.8\} \rightarrow \{p_{dog}^{full} = 0.45; p_{cat}^{full} = 0.55\} \rightarrow \cdots$$

… results to preferring the "dog" over the "cat" label for the original image, i.e.:

$$\ldots \rightarrow \{p_{dog}^{res} = 0.52; p_{cat}^{res} = 0.48\}.$$

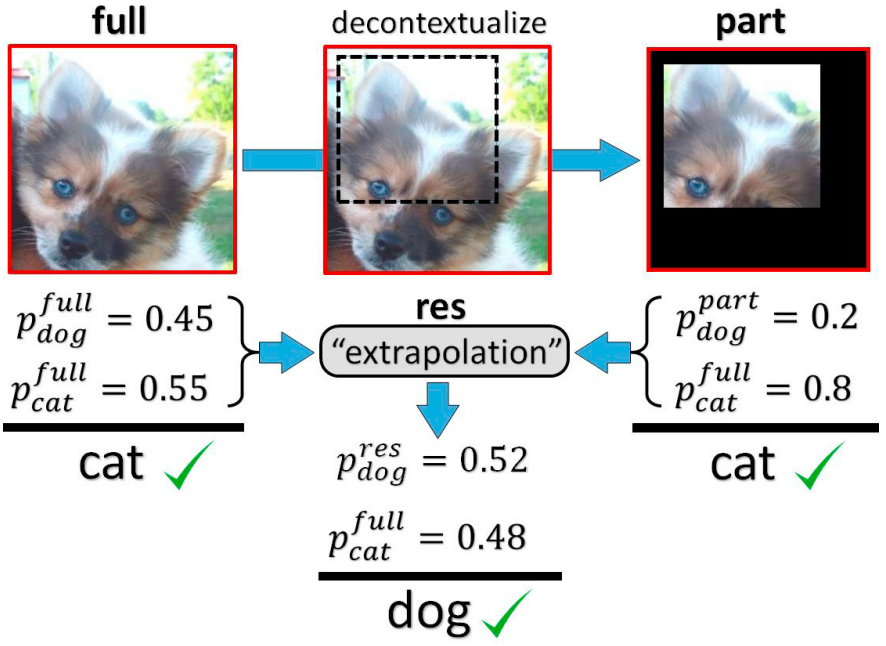

Fig. 1. Illustrating the intuition behind the "decontextualize-and-extrapolate" approach to classification. The input image "full" is classified separately (by some CNN) from its chosen and decontextualized "part". Results of classification for both "full" and "part" images give preference to the "cat" label. However, the study of the "part-to-full" tendencies (extrapolation) gives the grounds to assume that some potential and more informative (unknown, abstract) image "res" of the same object may have another classification label, i.e., "dog", which makes it reasonable to prefer the "dog" over the "cat" as a classification label also for the original image in spite of the classifier preference.

The intuition behind such a "decontextualize-and-extrapolate" approach is motivated by our former studies in the nineties [9] and [10], which suggested the way to handle uncertain information acquired from multiple sources. Each source is supposed to give its own interval estimation of the value of some parameter having different conditions for the estimation (noise, quality of sensors, etc.). The goal was to process all the intervals, discover the trends (i.e., in which direction the estimated value goes by increasing the quality of the estimation conditions) and derive the resulting estimation that is more precise than the original ones. Now, due to the recent advances in deep learning and particularly

with the growing popularity of CNNs for image processing, we are able to test and possibly benefit from our former approach to modern image classification problems.

Further questions (to be considered in the following sections) are as follows: how to define an appropriate part (one or several) of an image for decontextualization? How to measure the "quality" of an entire image and of its part, i.e. ω_{full} and ω_{part}? How to compute a better estimation of the class probability on the basis of the estimations provided by the classifier for the entire image and for its part? At which level of abstraction are we supposed to perform such "decontextualize-and-extrapolate" manipulations (at the "surface" or pixel level or going deeper into the features of the image and its part after the convolutional layers)?

3. An optimistic computational schema for the "just-one-part" classification refinement

Assume that we have some CNN classifier already trained on some image dataset to classify images into one of C classes. Assume that we are testing this classifier on a test image "full" taken entirely as an input. Let us also assume that, in this section, we will use the simplest estimate for the amount of information (ω_{full}) within an image, i.e., the number of pixels in it. During the classification process, the last SoftMax layer of the CNN classifier provides (as output) the computed probability distribution of the image belonging to each of C possible classes as follows:

$\{p_1^{full}, p_2^{full}, \dots, p_C^{full}\}$, where $\forall p_i^{full}\left(p_i^{full} \geq 0\right); p_1^{full} + p_2^{full} + \dots + p_C^{full} = 1.$ (1)

Normally, the class with the maximal value of probability in this vector is taken as a final label assigned to the input test image. We, however, are going to challenge this traditional way of defining the winner.

Assume that we cut some integral "part" of the input image "full" and decontextualize it (put zeros instead of the pixels outside the chosen part). The number of pixels within the "part" will be denoted as ω_{part}. If we apply the CNN classifier to the chosen part as a separate image, we will get another probability distribution as a result:

$\{p_1^{part}, p_2^{part}, \dots, p_C^{part}\}$, where $\forall p_i^{part}\left(p_i^{part} \geq 0\right); p_1^{part} + p_2^{part} + \dots + p_C^{part} = 1.$ (2)

Now, following the logic of the previous section, we assume that there could be some (yet unknown) abstract image (named "res") wider than the original "full" and such that:

$\omega_{full} = \frac{\omega_{part}+\omega_{res}}{2}$, which means that $\omega_{res} = 2 \cdot \omega_{full} - \omega_{part}$. (3)

Can we guess what would be the result of classification, if you apply the CNN classifier, taking the abstract "res" image as an input? Therefore, we want to estimate the following probability distribution:

$\{p_1^{res}, p_2^{res}, \dots, p_C^{res}\}$, where $\forall p_i^{res}(p_i^{res} \geq 0); p_1^{res} + p_2^{res} + \dots + p_C^{res} = 1.$ (4)

Let us suggest a simple computational schema of estimating distribution (4) given distributions (1) and (2) and assumption (3). The heuristic assumption here would be the following weighted average schema:

$p_i^{full} = \frac{\omega_{part} \cdot p_i^{part} + \omega_{res} \cdot \tilde{p}_i^{res}}{\omega_{part}+\omega_{res}}$, and, therefore, the preliminary (not normalized) estimate $\tilde{p}_i^{res}$ for p_i^{res} would be (taking (3) into account):

$\tilde{p}_i^{res} = \frac{2 \cdot \omega_{full} \cdot p_i^{full} - \omega_{part} \cdot p_i^{part}}{2 \cdot \omega_{full} - \omega_{part}}.$ (5)

Now we need to normalize (5) to guarantee that $\forall p_i^{res}(p_i^{res} \geq 0); p_1^{res} + p_2^{res} + \dots + p_C^{res} = 1$. We suggest using two different options to do that:

(A) The *SoftMax* option, which will use the SoftMax function, i.e.:

$p_i^{res} = \mathbf{SoftMax}(\tilde{p}_i^{res}) = \frac{e^{\tilde{p}_i^{res}}}{\sum_{j=1}^{C} \tilde{p}_j^{res}};$ (6)

Special cases of formula (6) for more comfortable computing are as follows:

$p_i^{res} = \mathbf{SoftMax}\left(\frac{2 \cdot \alpha \cdot p_i^{full} - p_i^{part}}{2 \cdot \alpha - 1}\right)$, if $\frac{\omega_{full}}{\omega_{part}} = \alpha;$ (6a)

$p_i^{res} = \mathbf{SoftMax}\left(\frac{4 \cdot p_i^{full} - p_i^{part}}{3}\right)$, if $\frac{\omega_{full}}{\omega_{part}} = 2;$ (6b)

$$p_i^{res} = \mathbf{SoftMax}\left(\frac{(1+\sqrt{5})\cdot p_i^{full} - p_i^{part}}{\sqrt{5}}\right), \text{ if } \frac{\omega_{full}}{\omega_{part}} = \varphi = \frac{1+\sqrt{5}}{2} \approx 1.618 \text{ - "Golden Ratio" constant.} \quad (6c)$$

(B) *Shift-and-normalize* (we name it as *basic* and computationally less expensive than the previous one) option, which is as follows:

$$p_i^{res} = \frac{2\cdot\omega_{full}\cdot p_i^{full} + \omega_{part}\cdot(1-p_i^{part})}{2\cdot\omega_{full} + \omega_{part}\cdot(C-1)}. \quad (7)$$

Special cases of formula (7) are as follows:

$$p_i^{res} = \frac{2\cdot\alpha\cdot p_i^{full} - p_i^{part} + 1}{2\cdot\alpha + C - 1}, \text{ if } \frac{\omega_{full}}{\omega_{part}} = \alpha; \quad (7a)$$

$$p_i^{res} = \frac{4\cdot p_i^{full} - p_i^{part} + 1}{3 + C}, \text{ if } \frac{\omega_{full}}{\omega_{part}} = 2; \quad (7b)$$

$$p_i^{res} = \frac{(1+\sqrt{5})\cdot p_i^{full} - p_i^{part} + 1}{\sqrt{5} + C}, \text{ if } \frac{\omega_{full}}{\omega_{part}} = \varphi = \frac{1+\sqrt{5}}{2} \approx 1.618 \text{ - "Golden Ratio" constant.} \quad (7c)$$

If, for example, we have the case $\frac{\omega_{full}}{\omega_{part}} = 2$ in Fig. 1, then by using the *SoftMax* computation option (6b) we get:

$\{p_{dog}^{full} = 0.45; p_{cat}^{full} = 0.55\}$ with $\{p_{dog}^{part} = 0.2; p_{cat}^{part} = 0.8\} \rightarrow \{p_{dog}^{res} = 0.51(6); p_{cat}^{res} = 0.48(3)\}$.

If to use the *basic* computation option (7b) for the same case then we have the result already shown in Fig. 1:

$\{p_{dog}^{full} = 0.45; p_{cat}^{full} = 0.55\}$ with $\{p_{dog}^{part} = 0.2; p_{cat}^{part} = 0.8\} \rightarrow \{p_{dog}^{res} = 0.52; p_{cat}^{res} = 0.48\}$.

We may see that in both cases the preference will be finally given to the "dog" label (which is the correct one for the original image) in spite of the original image classification by CNN.

The analytical options listed above could be used as part of the testing phase for the supervised learning tasks with CNNs as shown in Fig.2. One may see that both the entire image and its chosen part go through the same CNN and the two resulting probability distributions will be combined into a final one by the Decontextualization component.

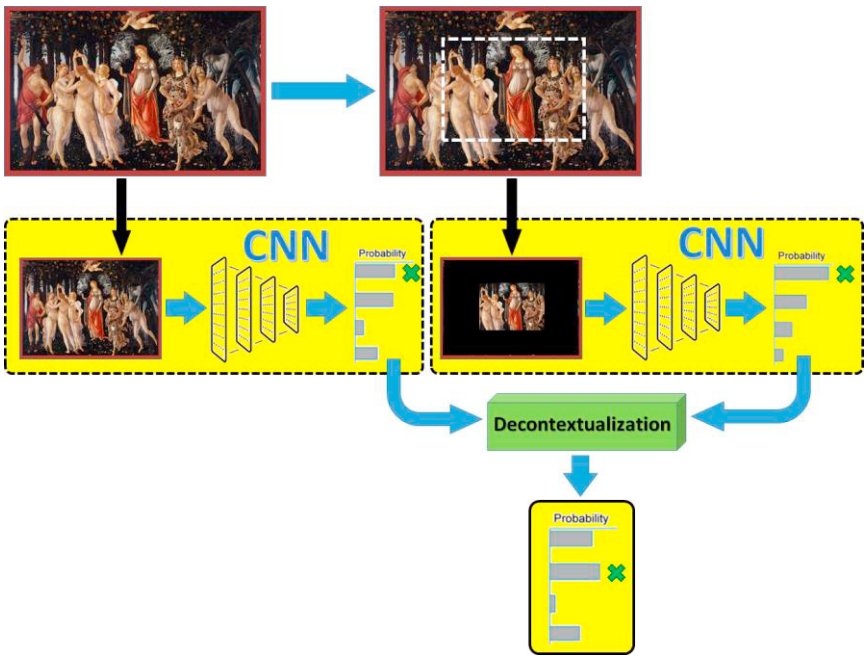

Fig. 2. Illustrating the generic schema of the simple option of the "decontextualize-and-extrapolate" approach. The entire input image and its chosen part go through the same CNN for classification. Both outcomes are integrated into one following the appropriate computation schema. Such refinement of the CNN classification outcome is expected to provide useful corrections and improve the classification accuracy on the test set.

These simple (naïve) schemas may give a certain improvement to classification accuracy. However, they depend a lot on the choice of the "part" of the analysed image. We are going to correct this in the following section.

4. Computational schema for the "sliding focus" classification refinement

Assume that we are working with the colored images of a size $[3 \times m \times n]$ in pixels according to the RGB schema. Assume that we want to use the "decontextualize-and-extrapolate" approach described above with such images. This would mean that the simplest estimate for the amount of information (ω_{full}) within each image will be as follows: $\omega_{full} = 3 \cdot m \cdot n$. Then, to be able to apply the above analytics, we need to define the size and location of the image "part" 3D frame $[3 \times \tilde{m} \times \tilde{n}]$, where $\tilde{m} < m, \tilde{n} < n$, and, therefore, $\omega_{part} = 3 \cdot \tilde{m} \cdot \tilde{n}$, $\frac{\omega_{full}}{\omega_{part}} = \alpha = \frac{m \cdot n}{\tilde{m} \cdot \tilde{n}}$. We recommend using α to be either 2 or φ ("Golden Ratio"). If we choose just one particular location of the "part" frame as a kind of focus in addition to the entire image (as, e.g., presented in Fig.1), then we will be biased towards this choice, and the correction computed on its basis cannot be fully trusted. Therefore, in this section, we consider a more computationally expensive yet more consistent way to apply the "decontextualize-and-extrapolate" approach to image classification.

The chosen frame (aka focus) will be a sliding 3D window, which will be moved step-by-step across the entire image. The size of each step (both horizontal and vertical) will be denoted as "stride". For each i-th "part", which happens to be within the frame during its sliding process, we get the classification result (probability distribution) and collect it into the following matrix:

$$P = \begin{Bmatrix} p_1^{part_1}, & p_2^{part_1}, \dots, & p_C^{part_1} \\ p_1^{part_2} & p_2^{part_2}, \dots, & p_C^{part_2} \\ \dots & \dots & \dots \\ p_1^{part_g} & p_2^{part_g}, \dots, & p_C^{part_g} \end{Bmatrix}, \quad (8)$$

where g is the number of different parts collected during the "sliding focus" process; each i-th row represents probability distribution as a result of classification of the i-th "part" of the original image; each j-th column represents probabilities (aka "votes") from each part regarding the choice of j-th class label for the image.

Now, using the values from matrix (8), we are able to compute more solid and unbiased value for each of the p_k^{part} component in the distribution from formula (2) as a simple average:

$$p_k^{part} = \frac{1}{g} \cdot \sum_{i=1}^{g} p_k^{part_i}. \quad (9)$$

After that we can apply any of the computational schemas from the previous section to get refined estimations p_k^{res} from p_k^{full} together with the more smartly computed p_k^{part}.

How to compute the number of parts g for the "sliding focus" method? This will depend on m, n, $\tilde{m}$, $\tilde{n}$ and stride as follows:

$$g = \left\lfloor \frac{(m-\tilde{m}) \cdot (n-\tilde{n})}{\text{stride}^2} \right\rfloor. \quad (10)$$

If, for example, we have some dataset with images of size $[3 \times m \times n] = 3 \times 128 \times 128$ and we want to consider the parts, which will be cut according to the "Golden Ratio" constant, we will get the following:

$$\omega_{full} = 3 \cdot 128 \cdot 128 = 49152;\ \omega_{part} = \frac{\omega_{full}}{\varphi} = \left\lfloor \frac{128}{\sqrt{\varphi}} \right\rfloor \cdot \left\lfloor \frac{128}{\sqrt{\varphi}} \right\rfloor \cdot 3 \approx 100 \times 100 \times 3 = 30000;\ \tilde{m} = \tilde{n} = 100.$$

Now, if to choose formula (10) and apply $\text{stride} = 4$, then we have:

$$g = \left\lfloor \frac{(128-100) \cdot (128-100)}{4^2} \right\rfloor = \left\lfloor \frac{784}{16} \right\rfloor = 49.$$

If in this example, the dataset has four categories (classes) of differently labelled images (i.e., $C = 4$) and we want to apply the *basic* computation schema (formula (7a)), then, by combining it with formula (9) we get the following formula for computing the refined probability distribution for our example:

$$p_k^{res} = \frac{160.5632 \cdot p_k^{full} + 49 - \sum_{i=1}^{49} p_k^{part_i}}{307.5632}.$$

Consider example in Fig. 3.

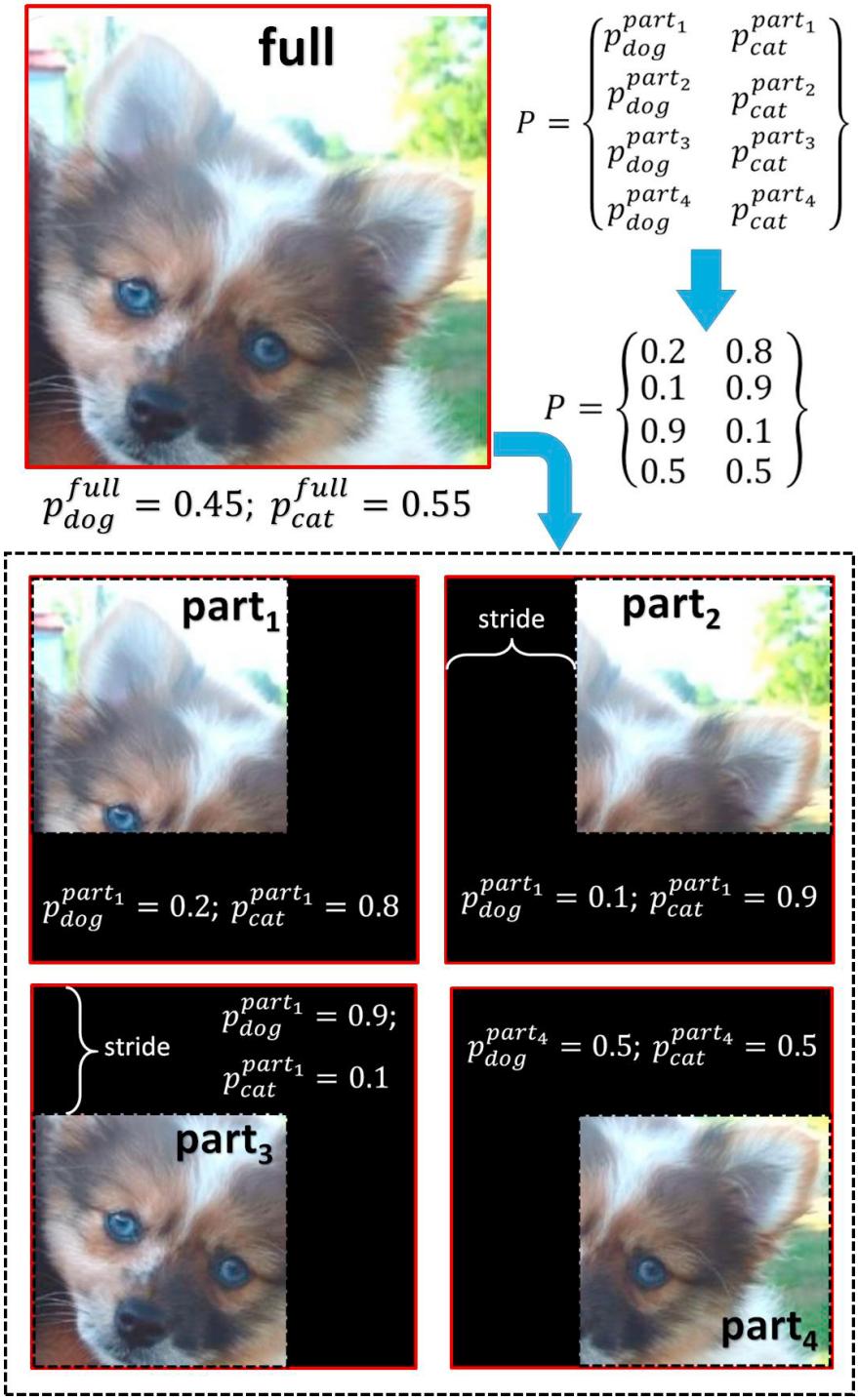

Fig. 3. Example of applying the "sliding focus" schema, which is intended to take into account several different parts (same size) of the original image to get an unbiased evaluation for the "part" component for the "decontextualize-and-extrapolate" refinement approach. In this example, the results of classification (got separately for each of the four parts of the original image) are collected into matrix *P*, which will be used for further refinement of the CNN classifier outcome obtained separately for the whole image.

The example illustrates the "sliding focus" procedure for the same image as the one in Fig.1. Here the size of the frame and the stride has been chosen so that we have only four options for the "part" to be considered. After the entire image and each of the four parts have been classified independently, we get all the needed components (which are: $\{p_{dog}^{full} = 0.45;\ p_{cat}^{full} = 0.55\}$ and matrix P as shown in Fig.3) for computing the refined probability distribution $\{p_{dog}^{res};\ p_{cat}^{res}\}$. In this example, we have $\alpha = \frac{\omega_{full}}{\omega_{part}} = 2.25$. According to formula (9), we get:

$$p_{dog}^{part} = \frac{1}{g} \cdot \sum_{i=1}^{g} p_{dog}^{part_i} = \frac{1}{4} \cdot (0.2 + 0.1 + 0.9 + 0.5) = 0.425; \text{ and}$$

$$p_{cat}^{part} = \frac{1}{g} \cdot \sum_{i=1}^{g} p_{cat}^{part_i} = \frac{1}{4} \cdot (0.8 + 0.9 + 0.1 + 0.5) = 0.575.$$

And, finally, by applying computational schema from formula (7a), we get the following refined probability distribution:

$$p_{dog}^{res} = \frac{2 \cdot \alpha \cdot p_{dog}^{full} - p_{dog}^{part} + 1}{2 \cdot \alpha + C - 1} = \frac{2 \cdot 2.25 \cdot 0.45 - 0.425 + 1}{2 \cdot 2.25 + 2 - 1} = \frac{4.5 \cdot 0.45 - 0.425 + 1}{5.5} = 0.4(72); \text{ and}$$

$$p_{cat}^{res} = \frac{2 \cdot \alpha \cdot p_{cat}^{full} - p_{cat}^{part} + 1}{2 \cdot \alpha + C - 1} = \frac{2 \cdot 2.25 \cdot 0.55 - 0.575 + 1}{2 \cdot 2.25 + 2 - 1} = \frac{4.5 \cdot 0.55 - 0.575 + 1}{5.5} = 0.5(27).$$

As one may see: the entire image classification by CNN has given us $\{p_{dog}^{full} = 0.45;\ p_{cat}^{full} = 0.55\}$ probability distribution, which means "cat" as a label for the image; the biased "decontextualize-and-extrapolate" approach (Fig.1) results in $\{p_{dog}^{full} = 0.52;\ p_{cat}^{full} = 0.48\}$ distribution and, therefore, "dog" label; and, finally, the unbiased (due to the "sliding focus") "decontextualize-and-extrapolate" approach (Fig.3) gives us $\{p_{dog}^{full} = 0.4(72);\ p_{cat}^{full} = 0.5(27)\}$ distribution. Knowing that the actual label for the image is "dog" (i.e., the distribution done by a "perfect" classifier must be $\{p_{dog}^{full} = 1;\ p_{cat}^{full} = 0\}$), we admit that the traditional classifier (applied on top of the entire image) has made a classification mistake; the naïve (biased) refinement has "guessed" the correct label for the image; and the unbiased and solid refinement still made a mistake but it happens to be about 2.3% smaller than the mistake from a traditional classifier. Therefore, with one image, the classification accuracy loss has been substantially decreased, which means that, in general, the suggested "sliding focus" refinement procedure on top of traditional CNNs could potentially improve image classification accuracy.

It would also be important to mention that, if the computational resource (particularly time) is not a problem, and we have high-resolution images, then all the above analytics could be recursively applied to the parts of the image the same way as to the entire image and, therefore, the analytics will produce smaller and smaller parts at each iteration and resulting to deeper refinement of the original image classification through the whole chain of refinements for the parts' classification.

5. Computational schemas for an entropy-aware refinement

In the previous sections, we used the number of pixels as a simple estimate for the amount of information in the images and their parts ($\omega_{full}, \omega_{part}$). In this section, we will adapt the computational schemas to a more solid measure of information (used in information theory), which is entropy. Assume that we are working with a colored image X (size $[3 \times m \times n]$ in pixels), which is represented by three $[m \times n]$ RGB channels X_1, X_2, X_3.

Claude Channon had defined information entropy already in 1948 [11] as a measure of the information content within a message (i.e., a measure of uncertainty reduced by the message). The general definition considers such a message (textual, image, etc.) as a random variable, which takes values from some "alphabet" (of v items) with a certain probability distribution of the items within the alphabet. Information entropy is defined as follows:

$$H(X) = -\sum_{i=1}^{v} [p(x_i) \cdot \log_2 p(x_i)]. \quad (11)$$

where the sum is taken over all variable's possible values.

It would be easy to adapt this definition to the case of image entropy. We have an "alphabet" of 256 intensity levels for the pixels. To get the needed probabilities we can use the histogram of an image [12]. To compute the entropy for the color image, we can average the entropies of each RGB channel computed separately. Therefore, the entropy of a

color image X will be as follows:

$$H(X) = \frac{1}{3} \cdot \sum_{r=1}^{3} \{ - \sum_{s=0}^{255} [p_s(X_r) \cdot \log_2 p_s(X_r)] \}, \quad (12)$$

where $p_s(X_r) = \frac{\text{Number of occurences of the intensity level } s \text{ within the } r^{\text{th}} \text{ RGB channel of image } X}{\text{Number of pixels (size) within the the } r^{\text{th}} \text{ RGB channel of image } X}$.

Once we know the way to compute an image entropy and, therefore, to have better estimates for ω_{full} and ω_{part}, we can adapt the previous computational schemas (just-one-part (section3) and "sliding focus" (section 4)) accordingly.

5.1. An entropy-aware "just-one-part" biased refinement schema

In spite of the fact that the optimistic "just-one-part" refinement schema (section 3) is obviously biased to the choice of a "part" or a cut from a classified image, entropy awareness provides the possibility to improve it. Before we were dealing just with volume of pixels, i.e., from the image of size $[3 \times m \times n]$ we randomly chose the part of size $[3 \times \tilde{m} \times \tilde{n}]$ so that we had $\frac{\omega_{full}}{\omega_{part}} = \frac{m \cdot n}{\tilde{m} \cdot \tilde{n}} = \alpha$ (e.g., $\alpha = 2$ for the "Dichotomy" case; or $\alpha = \varphi$ for the "Golden Ratio" case). Now we can update this approach towards having $\frac{\omega_{full}}{\omega_{part}} = \frac{H(\text{full})}{H(part)} = \alpha$ (e.g., $\alpha = 2$ for the "Dichotomy" case; or $\alpha = \varphi$ for the "Golden Ratio" case). Therefore, now we can formulate the task of finding "the best" part as follows: how to find the smallest possible "part" (3D frame or cut) of the original image so that it keeps the α-proportion above. This would mean that we are going to find the "part" with the highest density of information and with the fixed proportion of entropy with the entire ("full") image. A possible algorithm to do that could be as follows:

THE "BEST PART" SEARCH

INPUTS: original colored image X of $[3 \times m \times n]$ size; choice for the case {"Dichotomy" or "Golden Ratio"};

OBJECTIVES: weight ω_{full} for the original image; part $\tilde{X}$ of the original image X and its weight ω_{part}.

THE ALGORITHM:

Initialize: $\tilde{X}_0 = X$; $\tilde{X} = X$; $i = 0$;

Compute: $H(\tilde{X}_0)$ according to formula (12)

Assign: $\omega_{full} = H(\tilde{X}_0)$;

(*) Increment i;

i-th ITERATION LOOP:

- [remove the right column (in all 3 RGB channels) if possible from the $\tilde{X}_{i-1}$ and get $\tilde{X}_i^{right}$];
- [remove the left column (in all 3 RGB channels) if possible from the $\tilde{X}_{i-1}$ and get $\tilde{X}_i^{left}$];
- [remove the top row (in all 3 RGB channels) if possible from the $\tilde{X}_{i-1}$ and get $\tilde{X}_i^{top}$];
- [remove the bottom row (in all 3 RGB channels) if possible from the $\tilde{X}_{i-1}$ and get $\tilde{X}_i^{bottom}$];
- [for each of four $\tilde{X}_i^j$ compute $H(\tilde{X}_i^j)$ and entropy loss $H(\tilde{X}_{i-1}) - H(\tilde{X}_i^j)$];
- [from the four $\tilde{X}_i^j$ choose the one $\tilde{X}_i^{best}$ with the minimal entropy loss];
- [assign $\tilde{X}_i = \tilde{X}_i^{best}$; $H(\tilde{X}_i) = H(\tilde{X}_i^{best})$];

EITHER: CASE "Dichotomy": IF: $\frac{H(\tilde{X}_0)}{H(\tilde{X}_i)} \geq 2$, …

OR: CASE "Golden Ratio" : IF: $\frac{H(\tilde{X}_0)}{H(\tilde{X}_i)} \geq \frac{1+\sqrt{5}}{2} \approx 1.618$, …

… THEN (FOR BOTH CASES): assign $\tilde{X} = \tilde{X}_i$; $\omega_{part} = H(\tilde{X}_i)$ and STOP ALGORITHM;

- ELSE: continue to (*) …

After getting the intended "part" of the image and the entropy-based evaluations (ω_{full} and ω_{part}) for both: image and its part, we can use: either any of the cases for the *SoftMax* refinement option (formulas (6), (6a), (6b), and (6c)); or any of the cases for the *Basic* refinement option (formulas (7), (7a), (7b), and (7c)).

5.2. An entropy-aware "sliding focus" unbiased refinement schema

The "sliding focus" refinement schema (section 4) uses the same size frame to collect all intended parts of the original image to be used for further classification and $\forall i(\omega_{part_i} = \omega_{part})$. In the new entropy-awareness context, these weights (or corresponding entropies) could be different for different parts collected by the sliding frame. Therefore, we have to change a simple average formula (9) to a weighted average. The computational schema for the "sliding focus" refinement from section 4 is updated here by the formulas (13), (14) and (15) as follows:

$$\omega_{full} = H(\text{full}) = H(X); \tag{13}$$

$$p_k^{part} = \frac{\sum_{i=1}^{g} \omega_{part_i} \cdot p_k^{part_i}}{\sum_{i=1}^{g} p_k^{part_i}}, \text{ where } \omega_{part_i} = H(\text{part}_i); \tag{14}$$

$$\omega_{part} = \frac{1}{g} \cdot \sum_{i=1}^{g} \omega_{part_i}. \tag{15}$$

6. From pixels to features (deep "semantic" refinement schemas)

Similar schemas as discussed above, which are supposed to work on the level of pixel representation of an image, could be adapted also to the level of the features discovered from the image. It is known that convolutional layers of a CNN architecture discover the features of an input image with respect to its location in the image; and then these features are used by the rest (fully-connected layers) of the network to classify an image (compute probability distribution among the possible classes).

A generic schema of CNN capable of the "decontextualize-and-extrapolate" refinement at the level of features (i.e., semantic level) is illustrated in Fig. 4. One can see that, in addition to the normal information flow in the network, which uses all the discovered features for classification, there is a parallel information flow, which uses only particular cuts from the feature maps (i.e., decontextualized from the rest of the features "parts"). Both flows result in probability distributions p_k^{full} and p_k^{part}, which are used to compute the refined distribution p_k^{res} using analytics, which has already been described in the paper. The only (fortunate) specific here is that the feature maps already represent the values of the presence of particular features within image locations, and, therefore, just sums of values within the chosen frames can be used to measure ω_{full} and ω_{part} (i.e., no need for specific entropy computing).

7. Summary of the experiments

For fast check of the suggested analytics, we did image classification experiments with the public and popular image dataset Kaggle ("Cats and Dogs" Dataset: www.microsoft.com/en-us/download/details.aspx?id=54765), which contains 50K colored images of various cats and dogs. For fair comparison, we have not used any pre-trained models. We have trained the baseline CNN model and we have tested it without and with different refinement enhancements to compare various approaches.

Table 1 contains examples of our experiments. We may see that refinement improves classification in all the refinement cases, even for such a relatively simple dataset for ML. As expected, unbiased refinement performed a bit better than a biased one and the semantic (feature) refinement shows the best test accuracy. However, processing time, especially for entropy-aware computing, is essentially higher than for the baseline model (subject of optimization).

For making the final judgement about the hidden potential of the discussed "decontextualize-and-extrapolate" refinement techniques for improving image (and, possibly, not only image) classification, one would need many more experiments with different datasets. This is planned as an objective for our future studies.

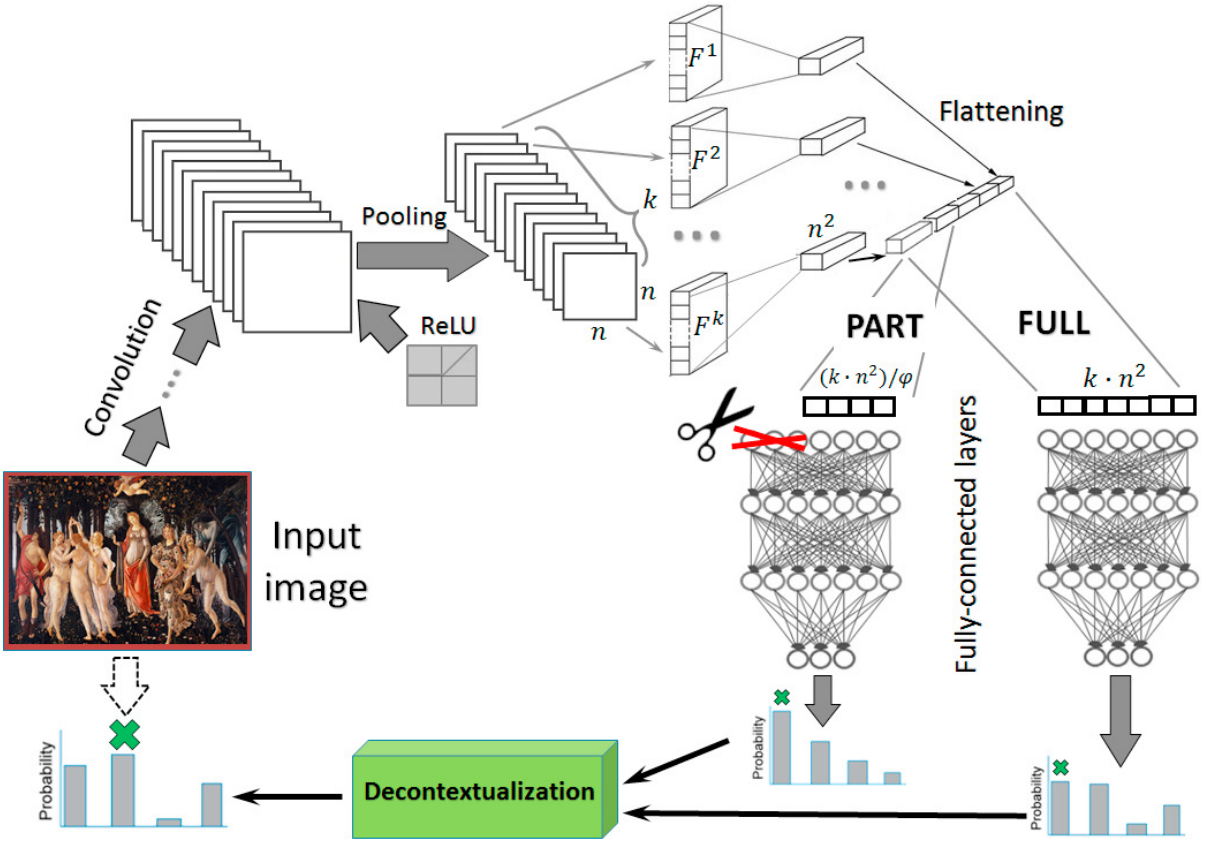

Fig. 4. A generic schema of "semantic refinement" (at the level of features) of image classification within a CNN architecture. The full discovered feature content (after convolutional layers) goes as usual through the fully-connected layers of the already trained CNN and gives one probability distribution as a result. The pruned feature content (after cutting away the features outside the chosen part(s)) goes through the same layers and gives another probability distribution as a result. Finally, both computed distributions go through the "Decontextualization" component of the analytics, which will compute the resulting and refined probability distribution as the final outcome of the input image classification.

Table 1. Comparisons of classification accuracy of CNN alone and several options of CNN + refinement over the Kaggle dataset images.

Baseline CNN model (without pre-training and without refinement) … + …	… + "Just-one-part" (SoftMax, "Golden Ratio").	… + "Just-one-part" (Basic, "Golden Ratio").	… + "Sliding Focus" (Basic, "Golden Ratio").	… + "Entropy Aware"/"Just-one-part" (Basic, "Golden Ratio").	… + "Entropy Aware"/"Sliding Focus" (Basic, "Golden Ratio").	… + "Semantic"/ "Sliding Focus" (Basic, "Golden Ratio").
82.11	83.02	83.19	84.55	83.98	84.71	85.34

8. Conclusions

Data-driven decision-making under uncertainty is an important concern of Industry 4.0 [13]. Image classification is an important topic within this agenda. In this paper, we suggested an approach ("decontextualize-and-extrapolate") to refine (improve) the outcomes of already trained ML classifiers, particularly of CNNs for image classification. We are talking here about images just with one object to be classified (recognized) and not about semantic segmentation (image clustering to capture different objects). We have based our analytics on the assumption that classification outcome for the image as a whole (i.e., getting a probability distribution among possible classes) will benefit from

classifying part(s) of the image separately (i.e., getting a probability distribution(s) among the same set of possible classes for the part(s)) and then combining these distributions in the extrapolation manner.

The intuition behind our approach is as follows: if you assume that Outcome (entire image) = FULL; Outcome (part of the image) = PART, then the refined classification result Outcome (Abstract image, wider than the original one) = RES, will not be a value between FULL and PART, but it can be obtained as extrapolation of these two. This means that our method assumes that any object to be classified does not contain all the information for perfect classification and, therefore: (a) there is no way to get more information from outside the object; we take some part(s) of the available information ("decontextualize" them from the rest of the image) to classify the same object even having less information available for that; (b) we discover the trend of the classification result change from less to more information; (c) finally, we use the trend to guess ("extrapolate") what would be the classification result if we have even more information that the original object has. Suggested refinement analytics provides different options for making such an extrapolation at different layers of information about the object to be classified: from the "surface" layer (e.g., pixels of an image) to the "semantic" layer (e.g., discovered feature maps from an image).

Our preliminary experiment on a public dataset provides certain optimism towards the validity of the suggested refinement techniques. However, many more experiments are still foreseen to discover the full hidden potential or the suggested approach. The nearest plan for our future research is to check the analytics with the datasets related to the security of the industrial logistics processes within the IMMUNE project [14].

References

[1] Longo, F., Nicoletti, L., and Padovano, A. (2019). "A system for supply chains diversification and (re) design: supporting managers' perspective in the face of uncertainty". *International Journal of Logistics Systems and Management*, **32(2)**: 168-194.

[2] Longo, F., Nicoletti, L., and Padovano, A. (2019). "Modeling workers' behavior: A human factors taxonomy and a fuzzy analysis in the case of industrial accidents". *International Journal of Industrial Ergonomics*, **69**: 29-47.

[3] Rai, R., Tiwari, M. K., Ivanov, D., and Dolgui, A. (2021). "Machine learning in manufacturing and industry 4.0 applications". *International Journal of Production Research*, **59(16)**: 4773–4778.

[4] Kläs, M., and Vollmer, A. M. (2018). "Uncertainty in machine learning applications: A practice-driven classification of uncertainty". In: *Proceeding of the International Conference on Computer Safety, Reliability, and Security* (pp. 431-438). Springer, Cham. https://doi.org/10.1007/978-3-319-99229-7_36

[5] Campagner, A., Cabitza, F., and Ciucci, D. (2020). "Three-way decision for handling uncertainty in machine learning: A narrative review". In: *Proceedings of the International Joint Conference on Rough Sets* (pp. 137-152). Springer. https://doi.org/10.1007/978-3-030-52705-1_10

[6] Terziyan, V., and Puuronen, S. (1999). "Knowledge acquisition based on semantic balance of internal and external knowledge". *Lecture Notes in Computer Science*, **1611**: 353-361. Springer. https://doi.org/10.1007/978-3-540-48765-4_39

[7] Arbeláez, P., Hariharan, B., Gu, C., Gupta, S., Bourdev, L., and Malik, J. (2012). "Semantic segmentation using regions and parts". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3378-3385). IEEE. https://doi.org/10.1109/CVPR.2012.6248077

[8] Jeczmionek, E., and Kowalski, P. A. (2021). "Flattening layer pruning in convolutional neural networks. *Symmetry*, **13(7)**: 1147. https://doi.org/10.3390/sym13071147

[9] Gerasin, S., Kaikova, H., and Terziyan, V. (1990). "An interval-based approach to multiple experts' opinions processing: successive severance method. *Bionics Problems Journal*, **44**: 41-46. https://openarchive.nure.ua/bitstream/document/12438/1/44_Gerasin_41-46.pdf

[10] Terziyan, V., Puuronen, S., and Kaykova, H. (1999). "Interval-based parameter recognition with the trends in multiple estimations". *Pattern Recognition and Image Analysis (Advances in Mathematical Theory and Applications)*, **9(4)**: 719-731. http://www.cs.jyu.fi/ai/papers/vt_sp_hk_PatternRecog99.pdf

[11] Shannon, C. E. (1948). "A mathematical theory of communication". *The Bell System Technical Journal*, **27(3)**: 379-423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

[12] Thum, C. (1984). "Measurement of the entropy of an image with application to image focusing". *Optica Acta: International Journal of Optics*, **31(2)**: 203-211. https://doi.org/10.1080/713821475

[13] Bousdekis, A., Lepenioti, K., Apostolou, D., and Mentzas, G. (2021). "A review of data-driven decision-making methods for industry 4.0 maintenance applications". *Electronics*, **10**(7): 828. https://doi.org/10.3390/electronics10070828

[14] Kaikova, O., Terziyan, V., Tiihonen, T., Golovianko, M., Gryshko, S., and Titova, L. (2022). "Hybrid threats against Industry 4.0: adversarial training of resilience". *E3S Web of Conferences,* **353**: 03004. https://doi.org/10.1051/e3sconf/202235303004