Double-View Matching Network for Few-Shot Learning to Classify Covid-19 in X-ray images

Gábor Szűcs[1] and Marcell Németh[2]

Abstract — **The research topic presented in this paper belongs to small training data problem in machine learning (especially in deep learning), it intends to help the work of those working in medicine by analyzing pathological X-ray recordings, using only very few images. This scenario is a particularly hot issue nowadays: how could a new disease for which only limited data are available be diagnosed using features of previous diseases? In this problem, so-called few-shot learning, the difficulty of the classification task is to learn the unique feature characteristics associated with the classes. Although there are solutions, but if the images come from different views, they will not handle these views well. We proposed an improved method, so-called Double-View Matching Network (DVMN based on the deep neural network), which solves the few-shot learning problem as well as the different views of the pathological recordings in the images. The main contribution of this is the convolutional neural network for feature extraction and handling the multi-view in image representation. Our method was tested in the classification of images showing unknown COVID-19 symptoms in an environment designed for learning a few samples, with prior meta-learning on images of other diseases only. The results show that DVMN reaches better accuracy on multi-view dataset than simple Matching Network without multi-view handling.**

Index Terms — **COVID-19, convolutional neural network, deep learning, feature extraction, few-shot learning, image classification, image representation, machine learning, multi-view**

I. INTRODUCTION

A necessary, but not sufficient condition for the effective use of machine learning (especially deep learning) methods is the availability of large amounts of training data. This condition cannot be satisfied in many applications (e.g., in image classification [15], especially in medical images [12]), in most cases due to a lack of available knowledge or excessive costs of expertise [23]. The research topic presented in this paper belongs to this problem type that is often lacking in such data, it intends to help the work of those working in medicine by analyzing pathological recordings, using only very few images. This scenario is a particularly hot issue nowadays: how could a new disease for which only limited data are available be diagnosed using features of previous diseases? (If the number of labeled data is small, but the huge amount of unlabeled data is available, then this can lead to active learning [14], but in this paper, we consider that there is no unlabeled data at all.)

[1] Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics (BME), Budapest, Hungary (e-mail: szucs@tmit.bme.hu)

[2] M. Németh is a student in BME Balatonfüred Student Research Group (e-mail: marcell.nemeth@edu.bme.hu)

In the case of learning from a small amount of labeled data, so-called "few-shot learning" (FSL), there are only a few samples from each class, so the difficulty of the classification task is to learn the unique feature characteristics associated with the classes as quickly and accurately as possible. In the few-shot learning literature *A-way-B-shot* means that we use B samples from A different classes for learning, so the training set has a total of $I = A \cdot B$ samples. This type of meta-learning in image recognition area requires knowledge transfer of high-level characteristics of training images similar to the target images. Although there are methods that can solve the problem, but if the images come from different views [17][24], they will not handle these views well. In this paper, we proposed an improved method, so-called Double-View Matching Network, which solves the few-shot learning problem as well as the different views of the pathological recordings in the images.

The next section discusses the theory of few-shot type machine learning in hypothesis space and its limits in Hilbert spaces. Then the paper presents the advanced methods, particularly the Matching Network (with a special focus on the attention mechanism and neural network architecture) for few-shot learning. For improvement and handling more views in the images, we suggested Double-View Matching Network, which is capable of recognizing multi-view recordings. The suggested method was tested in the classification of images showing unknown COVID-19 symptoms in an environment designed for learning a few samples, with prior meta-learning on images of other diseases only. The results of the new method are detailed at the end of the paper.

II. FEW-SHOT HYPOTHESIS LEARNING

A. Hypothesis Learning

Most few-shot learning tasks can be traced back to supervised learning problems, with only a few labeled samples from each class available to the classifier [4] (at unsupervised case, e.g., the graph clustering can be used [28]). The most common applications are image recognition, emotion recognition, object classification and multimedia analysis.

The general task of the problem is to parameterize a classifier h using only a very small number of samples that predicts label y_i for each input x_i. When a machine learner is trained on a large amount of training data, several models can be created at the end of the learning that are able to produce output from the input samples. However, with only a few data, a much larger number of such models can be "fitted" to the input-output pairs due to the wide variety of options (fewer constraints). These

DOI: 10.36244/ICJ.2021.1.4

models can be considered as hypothesis, that is, a function that produces the output from the input; and the aim is to find the best solution in this hypothesis space, as we present in the following based on a tutorial [10].

There is a function $f: X \to Y$, which can be quantified by the so-called empirical error:

$$R_{emp}[f] = \frac{1}{N}\sum_{i=1}^{N} L(f(x_i), y_i) \tag{1}$$

Thus, with the previous notations, we can formalize the problem of learning in an X input and Y output space, where D is an unknown distribution in an $X \times Y$ space and F is the hypothesis space for the functions

$$f : \mathrm{X} \to \mathrm{Y} \tag{2}$$

and $S = (x_1, y_1), \ldots, (x_N, y_N)$ samples from D. Based on these, the goal is to find a hypothesis $f \in F$ for which the real error is minimal:

$$R[f] = \mathbb{E}_D[L(f(x), y)] \tag{3}$$

The main disadvantage of the above relation is that it is not possible to minimize it clearly since we do not know the distribution D. However, it is possible to find an estimation by taking advantage of the fact that in most cases, the real error of the hypotheses takes on values significantly similar to the empirical error. The difference between the two errors is influenced by the flexibility of the used model (i.e., how many degrees of freedom it has). The disadvantage of a large number of degrees of freedom is that the hypothesis space is accompanied by a tendency to overfit the model, as we can fit innumerable functions to the desired distribution. Based on this idea, the so-called uniform convergence bounds can be defined, for all hypotheses *f* in a given hypothesis space, it is true that [10]:

$$R[f] \leq R_{emp}[f] + \varepsilon \tag{4}$$

where ε is the generalization error.

Despite the uniform convergence bounds defined by (4), there may be some training sets for which the model produces poor results, so the probability of the good results can be written with the following inequality for a given distribution D:

$$\mathbb{P}\left[R[f] - R_{emp}[f] \leq \varepsilon \mid \forall f \in F\right] \geq 1 - \delta \tag{5}$$

The main difficulty of finding a solution to this problem is that in the absence of accurate knowledge of D, the above relation must exist for all possible distributions of D in $X \times Y$ space (i.e., not just for a given distribution of D). However, inequality (5) should also be satisfied with a probability of $1 - \delta$ simultaneously for all hypotheses, thus for each function *f* we can write this formula within all possible distributions of D, so that we get the following:

$$\mathbb{P}\left[R[f] - R_{emp}[f] \leq \varepsilon\right] \geq 1 - \delta \quad \forall f \in F \tag{6}$$

The latter inequality expresses that for any given $f \in F$, except for the δ proportion of samples sampled "unlucky", equation (4) will be true. Inequality (6) is easier to accomplish because it is easier to achieve the same success rate in the whole set than the same rate within each subset. In contrast, the inequality (5) has the advantage that we can tell from sampling whether a given set of the training set is "lucky" or "unlucky". If it is "lucky", the inequality will be true for all hypotheses at once, i.e., we have achieved our goal. Based on this, it is advisable to write equation (5) in the following formula:

$$\mathbb{P}\left[\sup_{f \epsilon F}[\, R[f] - R_{emp}[f]\,] \leq \varepsilon\right] \geq 1 - \delta \tag{7}$$

The distinction between (7) and (6) is essential for what we want to use the bounds in the future; in the case of the FSL learning problem, the most important is the error of the hypothesis f^* for which the empirical error is minimal, which depends significantly on the choice of the training set.

The error of the model is influenced by the number of samples in the available training set S and the hypothesis space F. Starting from this statement, error minimization can be approached from several sides to reduce estimation inaccuracy using prior knowledge [26]. The possible approaches are the number of samples (a larger training set could help, but in FSL, only very few samples are available), and the algorithm for finding optimal parameters. The last method approaches the part of the model, which is responsible for defining and narrowing the hypothesis space. In this case, the use of a priori knowledge is aimed at reducing the complexity of the hypothesis space, excluding several potential hypotheses in advance.

B. Hilbert-space methods

Minimizing only empirical error is not sufficient, as this type of approach can lead to overfitting. To avoid this, it is necessary to narrow hypothesis F with certain limits. To solve this, starting from equation (4), we can introduce a penalty term, $\Omega[f]$, which quantifies the complexity of each hypothesis and minimizes the following error instead of the method presented in equation (1) [10]:

$$R_{reg}[f] = R_{emp}[f] + \Omega[f] \tag{8}$$

where $\Omega[f]$ is the regularization term, and R_{reg} is the regularized error. The learning problem should therefore focus on three components: the loss function L, the regularization term Ω and hypothesis space F.

In constructing the hypothesis space F, the natural expectation is that F is a linear function space in which for any $f \in F$ and λ, the product $\lambda \cdot f$ is also in F, and for any $f_1, f_2 \in F$ it is true that $f_1 + f_2 \in F$.

In addition, the structure of F should be related to the regularization term Ω in some way. This property is defined by an $\Omega[f] = \|f\|^2$ norms. For the new norm, the linear mappings taken with λ should also be satisfied and in order to obtain as a scalar product, let $\|f\| = \langle f, f\rangle^{1/2}$. These types of spaces are called Hibert space, and the great advantage of this is that Frigyes Riesz's theory can be applied to the present problem because in Hilbert spaces, the Riesz representation theorem is

true [6], as a result of which for any $x \in X$ there exists a representation k_x, for which it is true that:

$$f(x) = \langle k_x, f \rangle \quad \forall f \in F \tag{9}$$

We do not know k_x in equation (9), but we know for sure that it exists. A key element of the idea is that it creates a connection between the abstract structure of F and the elements in it, and we can use its representation instead of any x. If we rewrite the complete regularized error problem as follows

$$\hat{f} = \arg\min_{f \in F} \left[\frac{1}{N} \sum_{i=1}^{N} L\big(\langle k_{x_i}, f \rangle, y_i\big) + \langle f, f \rangle \right] \tag{10}$$

then it can be seen that f appears only in the form F with scalar products of other functions. It follows that if we know the scalar product and k_x, we will apply equation (9) with k_x to some $x' \in X$: $k_{x'}(x) = \langle k_x, k_{x'} \rangle = k_x(x')$. From the context, it can be seen that the inner products of the different $k_{x'}$ tell us what form each vector takes, and leaving the unnecessary elements, it can be seen that the quality of the algorithm is determined by the internal products called kernels:

$$k(x, x') = \langle k_x, k_{x'} \rangle \tag{11}$$

The only condition for kernels is that they should be symmetric as well as satisfying the following expression:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0, \tag{12}$$

where c_i, c_j are real coefficients, and they should result in $\langle \sum_{i=1}^{n} c_i k_{x_i}, \sum_{j=1}^{n} c_j k_{x_j} \rangle \geq 0$.

Summarizing the above, the learning problem has been traced back to the proper definition of an L loss function and k kernels. Looking at equation (10), it can be seen that $\hat{f}$ will be derived from the representatives of the training data $k_{x_1} \ldots k_{x_N}$, since the error term depends only on the internal product of f with different k, while the regularization term will affect all its dimensions. If f has a component that is orthogonal to the subspace spanned by $k_{x_1} \ldots k_{x_N}$, then the error term will not be affected, but the regularization term will be. It follows that the optimal f will entirely be in the subspace spanned by the representatives:

$$\hat{f}(x) = b + \sum_{i=1}^{N} \alpha_i k(x_i, x) \tag{13}$$

where $\alpha_1 \ldots \alpha_N$ are real coefficients and b is the offset (bias). Substituting formula (13) into (10), it can be observed that the task of the learning algorithm has been simplified to calculating the offset and the coefficients (the interpretation of these coefficients is the learnt knowledge after the learning).

III. MATCHING NETWORK ARCHITECTURE

A. Matching Network for few-shot learning

Several methods have been developed to solve the FSL problem: Prototypical Network [20], Attentive Recurrent Comparators [18], Simple Neural AttentIve Learner (SNAIL) [13], Memory-Augmented Neural Network (MANN) [2], ModelAgnostic Meta-Learning [3], Relation Network [21] and Siamese networks [9][16]. Based on the sources in the FSL literature, analyzing the results and considering further potential improvements, we chose one of the best methods, Matching Network [25], as the basis of our research. This solution adapts many techniques, including deep parameterized networks and metric learning [7] using feature vectors and deep neural networks with memory.

The essential idea of the Matching Network classifiers is to combine two learning phases: metric learning and the "lazy-learner" k-NN (k Nearest Neighbor) method. Metric learning is realized in the Hilbert-type spaces detailed in the previous section, while the k-NN-type classification takes place in the last phase during comparing feature vectors.

In the first phase, neural networks can be used. The main task of this phase is to learn a distance metric, a metric space in which the representations of samples from different classes are separated from each other as much as possible. Thus, the task of the applied neural networks is to parameterize a metric with such properties, i.e., an optimal hypothesis function to calculate the coefficients based on what is described in the previous section.

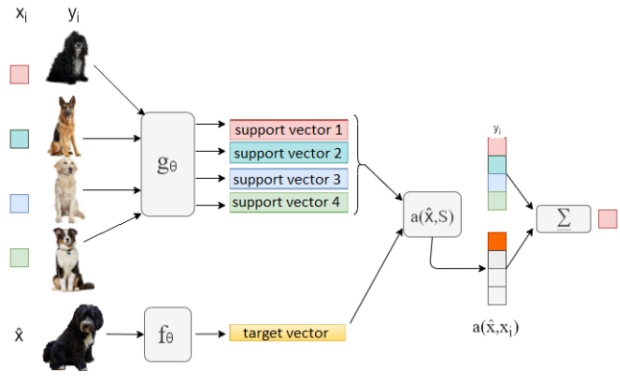

Figure 1. Matching Network architecture

The training set of the few-shot learner is called support set. The applied method in Matching Network defines a classifier ($S \rightarrow C_S(\cdot)$ mapping) for each support set sampled from the training set, and then combines the stored mappings to make the best use of the available knowledge. Thus, Matching Network type classifiers are able to categorize unknown classes with high efficiency without changing the networks.

In the following, we define the task description in more detail. Let S be a support set containing n sample-label pairs: $S = \{(x_i, y_i)\}_{i=1}^{n}$. As shown in Figure 1, the operation of the model was illustrated by recognizing dog breeds. The sample-label pairs (label means class label) of the support set are given as input to a classifier $C_S(\hat{x})$, which defines a probability distribution for a given sample $\hat{x}$ based on the class label $\hat{y}$. This mapping can be written as follows: $S \rightarrow C_S(\hat{x}) = P(\hat{y} \,|\hat{x}, S)$, where the function P is parameterized by neural networks. This construction allows us to use the model parameterized during learning to classify all elements of an S' support set containing unseen patterns. The class prediction of each sample can be

described as follows:

$$P(\hat{y}\,|\hat{x},S) = \sum_{i=1}^{n} a(\hat{x},x_i)y_i \tag{14}$$

where x_i, y_i are the samples, and their associated labels from the $S = \{(x_i, y_i)\}_{i=1}^{n}$ support set, and $a(\cdot,\cdot)$ is the kernel (also known as attention kernel or attention mechanism). It is worth noting that the above relation produces the output (label) of the samples of the new classes as a linear combination of the sample labels in the support set.

Appropriate selection of the model components that make up the attention kernel is key to the effectiveness of the model. In its most basic form, the kernel can be written using the softmax function applied to cosine distances as follows:

$$a(\hat{x},x_i) = \frac{e^{c(f(\hat{x}),g(x_i))}}{\sum_{j=1}^{n} e^{c\left(f(\hat{x}),g(x_j)\right)}} \tag{15}$$

where function c describes the cosine similarity, functions f and g are the neural networks responsible for constructing the feature vector formed from x_i and x (f and g preferably have the same architecture).

B. Multiple views challenge in images

In the first phase of the Matching Network, the appropriate feature vectors are essential. In our task, images of pathological chest X-rays were available; and we used a deep neural network (Convolutional Neural Network – CNN) detailed later to generate the characteristic vectors of the X-rays. The largest challenge in the image data set was that two types of recording perspectives - frontal and profile (side) recording - were stored for each disease type in the data set, but only one of them (frontal or profile) was available at each patient. In order to handle more views in the method, we developed an extended (improvement) version of the Matching Network, the so-called Double-View Matching Network. The next section presents this proposed method.

IV. DOUBLE-VIEW MATCHING NETWORK

A. Convolutional Neural Network for Feature Extraction

At image recognition topic, there is a frequent case that samples come from different perspectives. In the investigated dataset of medical images (in our case chest X-ray images), this was also true, the dataset contained two views. Our research focused on how recordings from the same class but from different perspectives can be used effectively. Our proposed method, the so-called Double-View Matching Network (DVMN for short) answers the question. In this section, we present the DVMN in two parts; firstly, the architecture and the details of the Convolutional Neural Network for feature extraction, then the solution of the combination of more views.

Proper selection of neural networks generating mappings from image into a common feature space (i.e., the feature extraction), is a key component of the accuracy. The publication of Matching Network [25], which is considered as the basic paper of our research, shared only small information about the neural network architecture for feature extraction that VGG [19] and Inception [22] networks can be used. However, these network architectures are not dedicated to medical images. Thus, we deviated from this approach and used our own structure, which is shown in Table 1, where each Convolution row consists of a convolution layer, then a batch normalization, and ReLu.

Images sampled from the set of training data serve as input to the convolutional network that produces the mapping. During the learning, an extra FC (fully connected) layer was added after the last layers of the CNN network to generate the output vectors. The CNN was used to the two networks, *f* and *g* having the same architecture (Figure 1.).

Operation layer	# filters	Size of filter	Stride value	Padding value	Size of output
Convolution	64	3x3x64	1x1	1x1	460x460x64
MaxPooling	1	2x2	2x2	0	230x230x64
Convolution	64	3x3x64	1x1	1x1	230x230x64
MaxPooling	1	2x2	2x2	0	115x115x64
Convolution	64	3x3x64	1x1	1x1	115x115x64
MaxPooling	1	2x2	2x2	0	57x57x64
Convolution	64	3x3x64	1x1	1x1	57x57x64
MaxPooling	1	2x2	2x2	0	28x28x64

Table 1. CNN network architecture

It is important to note that the mapping of each x_i per support set is independent of other samples. If the mapping of a sample x_i and x_j is close to each other in the parameter space, it is worthwhile to change the parameters of the model in order to refine the feature vectors, taking into account the mappings of other samples. Based on this idea, a component containing memory, the context embedding layer, was added to the network, similar to the original paper of Matching Network [25]. A bidirectional LSTM layer was used to embed each x_i sample, which stores the other feature mappings of the x_i sample support set:

$$f(\hat{x},S) = LSTM(f'(\hat{x}),g(S),K) \tag{17}$$

where $f'(\hat{x})$ denotes the characteristics generated by CNN that serve as input to the LSTM, $g(S)$ is the mapping of the given support set by g, and K is the number of "time steps" of the LSTM. This allows the attention mechanism to utilize only certain elements of the support set that add meaningful value to the mappings.

Context embedding of the classifier's f network based on equation (17) assuming a previous step k:

$$\hat{h}_k, c_k = LSTM(f'(\hat{x}),[h_{k-1},r_{k-1}],c_{k-1}) \tag{18}$$

$$h_k = \hat{h}_k + f'(\hat{x}) \tag{19}$$

$$r_k = \sum_{j=1}^{|S|} a\left(h_{k-1},g(x_j)\right)g(x_j) \tag{20}$$

$$a\left(h_{k-1},g(x_j)\right) = \frac{e^{h_{k-1}^{T}g(x_j)}}{\sum_{j=1}^{|S|} h_{k-1}^{T}\,g(x_j)}, \tag{21}$$

where x is the input, h is the output (cell after the output gate) and c is the memory cell. Furthermore, it is a function of the

attention mechanism with softmax activation. Context embedding of the classifier's g (target image) network:

$$g(x_l, S) = \overrightarrow{h_l} + \overleftarrow{h_l} + g'(x_l) \quad (22)$$

B. Handling the views

Our idea was to separate the different perspectives in some way in order to find a better model. In designing our solution with more views, the most important task was the optimal use of the feature vectors (hereafter vectors) of the images taken from each view. In DVMN, we suggested that the vectors of each perspective should be constructed by different CNN networks with separate parameters instead of a common one [5][27]. Behind this idea was that during training iterations, due to the small number of samples, tuning the model parameters to the appropriate "direction" is key, and recordings from different perspectives can easily miscalibrate weight settings. In addition, the mappings generated by the two separate networks need to be aggregated before classification [8], as the Matching Network would learn the difference between views instead of similarities among images from the same class, so our solution was based on the basic idea of working with a union of views.

Continuing the previous thoughts, let S_{L1} be a labeled image set that contains only the images in the first view and whose images we want to use to teach a metric space. To generate feature vectors from the images, we used a self-made CNN, because it is more flexible to learn new types of images than a pretrained deep neural network. By separating the last FC layers of CNN, the remained network generates a feature vector of n elements for each input image, denoting this feature extraction mesh as a function of f: $v_{L1} = f_{CNN1}(x)$. For all images in the tagged image set S_{L1}, the set of feature vectors generated in this way is denoted by V_{L1}:

$$V_{L1} = \{v_{L1} | v_{L1} = f_{CNN1}(x), x \in S_{L1}\} \quad (23)$$

The Matching Network generates a new vector from each entered feature vector that already describes the image in the new vector space, denoting this new vector by v'_{L1}, so that we can write that $v'_{L1} = f_{MN1}(v_{L1})$. The set of new vectors thus obtained is denoted by V'_{L1}:

$$V'_{L1} = \{v'_{L1} | v'_{L1} = f_{MN1}(v_{L1})\} \quad (24)$$

For an unknown class set (by unknown, we mean the set of classes belonging to the previous image set S_{L1} and the set of classes of the unknown set are disjoint sets, i.e., their intersection is an empty set, but the new set has some class labeled images) we want to use the learned new vector space, where the image set also consists only of images from the first view. With the previously learned CNN and MN models, vectors can be generated for all images (without the labels of the unknown image set), so we denote the set of new vectors obtained for the unknown image set by V'_{U1}, which will be:

$$V'_{U1} = \{v'_{U1} | v'_{U1} = f_{MN1}(f_{CNN1}(x)), x \in S_{U1}\} \quad (25)$$

If we select the vectors with the class label from the elements of V'_{U1} into the support set (this is the training set of the few-shot learner), we will be able to classify each of the other vectors with unknown class labels by predicting the class label whose the support vector is closest to the vector to be classified.

Using the notations used in the previous paragraphs in an analogous way to the second view:

$$V_{L2} = \{v_{L2} | v_{L2} = f_{CNN2}(x), x \in S_{L2}\} \quad (26)$$

$$V'_{L2} = \{v'_{L2} | v'_{L2} = f_{MN2}(v_{L2})\} \quad (27)$$

$$V'_{U2} = \{v'_{U2} | v'_{U2} = f_{MN2}(f_{CNN2}(x)), x \in S_{U2}\} \quad (28)$$

This mathematical framework of handling multi-view data is our contribution in this paper. In B-shot learning the images of each view are fed to the two CNNs, they will output two m long vectors. Let n_1 and n_2 be the number of samples in the first and in the second view, respectively in a given class. If the image dataset is ideal (that is $n_1 = n_2 = n$), the input data table will have a dimension $n \times 2m$ at the case of the concatenation of vectors belonging to two views. In a real environment, expecting an ideal dataset would be an unrealistic requirement, so the following options are available at this point:

- If at least one image is available from both views, but the number of images in a view is larger, the images already used can be re-input as the replacement for the missing images (in order to get the same number in each view). This method can easily lead to overfitting due to the repetition of samples.
- In order to get the same number in each view the other solution is the selection the minimum number among different views. In this case, a sample $min\,(n_1, n_2)$ is used from both views, so the size of the input data table will be $min(n_1, n_2) \times 2m$ at the case of concatenation. The negative result of this solution is the artificial reduction of the number and the expected decrease in accuracy based on the measurements.
- Instead of the concatenation of the vectors, we can get the union of the set of vectors. There is a requirement for the number of samples per view, the only condition is that $k_1 + k_2 \geq B$ (in B-shot learning). This solution with the union of views eliminates the imbalanced problem, thus our method works with this, and the dimension of the data table will be $(n_1 + n_2) \times m$.

C. DVMN on multiple views

During the DVMN method, we trained two CNNs based on the idea of a union of sets of vectors. Let V_{L1} and V_{L2} be sets of characteristic vectors analogous to equation (23) and (26). The solution presented below builds the model to take advantage of the union of views. Consider the union of feature vectors:

$$V_L = V_{L1} \cup V_{L2} \quad (29)$$

This complete set is given to the Matching Network (MN) to perform the vector space teaching required for a few-shot classification. The set of new vectors of $(n_1 + n_2) \times m$ thus obtained is denoted by V'_L:

$$V'_L = \{v'_L | v'_L = f_{MN}(v_L), v_L \in V_L\} \quad (30)$$

For using the learned new vector space for an unknown image set, the previously learned CNN1 and CNN2 (depending on whether the unknown image is in the first or second view) and MN can be applied to generate vectors for all images, so we denote the set of new vectors obtained for the unknown image

set by V'_U, which will be:

$$V'_U = \left\{ v'_U \middle| \begin{matrix} v'_U = f_{MN}(f_{CNN1}(x)), x \in S_{U1} \\ v'_U = f_{MN}(f_{CNN2}(x)), x \in S_{U2} \end{matrix} \right\} \quad (31)$$

The depicting part of the support set of the Double-View Matching Network architecture, which makes efficient use of multiple views and can handle the problem of unbalanced classes, is shown in Figure 2.

As a concluding point of the section, although the present implementation (described above) uses only two types of views due to the characteristics of the data set (and the architecture of the model), it would be able to take advantage of more different perspectives instead of two.

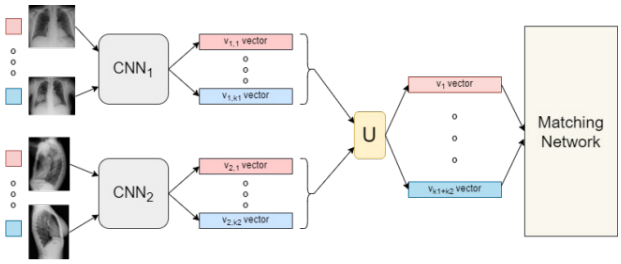

Figure 2. Mapping the Double-View Matching Network support vector

D. Training setup

So far, the operation of the Matching Network has been presented, which uses a set of support as an input of an $S \rightarrow C(x)$ classifier. In the method, using set-by-set sampling, a mapping in the form $P_\theta(y|\hat{x}, S)$ is obtained, in which θ denotes the parameters of the model.

In learning, in each iteration/epoch in which the gradients are calculated, and the model parameters are updated, we first sample a class set C from a set F (all classes) that contains a subset of all classes. Next, using C, we select the elements of the support set S, along with a S_B batch that contains some instances of the classes of the C set.

The model parameters are then parameterized in such a way that the error of the class predictions given to the samples in S_B is minimized when trained on S:

$$\theta = arg \max_\theta E_{C \sim F} \left[E_{S \sim C, S_B \sim C} \left[\sum_{(x,y) \in S_B} \log P_\theta(y|x, S) \right] \right] \quad (32)$$

Sampling batches through different iterations helps to avoid overfitting by providing the model with inputs of combinations of available images that it has not yet encountered in a given order of occurrence. This type of approach is particularly advantageous in context embedding, as identical sequences of repetitive images may in themselves lead to overfitting due to their repetitive (non-random) order through iterations. On the other hand, if we vary not only the images but also the order of their context learning, then the context embedding layer can perform different parameter tunings due to the changing environment.

V. EXPERIMENTAL RESULTS

A. Dataset

We investigated a real problem for testing the Double-View Matching Network method in the recognition and classification of viral diseases using pathological chest X-rays, for which only a very limited number of training samples are available. It is easy to imagine the potential of such a solution, which can detect new, almost unknown diseases without the use of extensive data collection and expertise (oracle), even at the beginning of an outbreak like COVID-19.

A publicly available COVID-19 data set [1] was used to design the experimental environment, which was expanded with chest X-rays of other diseases. It is important to note that the recordings are not assignable to humans, are completely anonymized, and do not contain any patient-specific information in their metadata. As we mentioned before, two types of recording perspectives for each disease in the data set are available - frontal and profile (side) recording. The well-known multi-view solutions [11] cannot be used because only one view was recorded at each patient.

The complete data set contains 758 records of a total of 19 disease classes. After data cleaning (removal of erroneous, watermarked, etc. recordings), 680 recordings from 15 classes were added to the final data set. Another special circumstance is that image collections from different sources have different resolutions, with the smallest reaching only 150x150 and the largest reaching up to 2500x2500 pixels. Regarding the classes of the data set, it shows an unequal number of samples for each perspective:

- Completely unbalanced (only images from one perspective can be found in the samples) classes include the following diseases: ecoli, ards, sars.
- Balanced (same number of images from both views): influenza, mycoplasma, bacterial, chlamydophila, COVID-19.
- For the other classes, there are recordings from both perspectives, but not in equal numbers: klebsiella, legionella, lipoid, pneumocystis, pneumonia, streptococcus, varicella.

The abnormalities of the lung caused by the COVID-19 virus are well recognized in the images in Figure 3 as a good example. Left side: symptoms are recognizable from "denser" lung areas, right side: "denser" areas are depicted on a heat map.

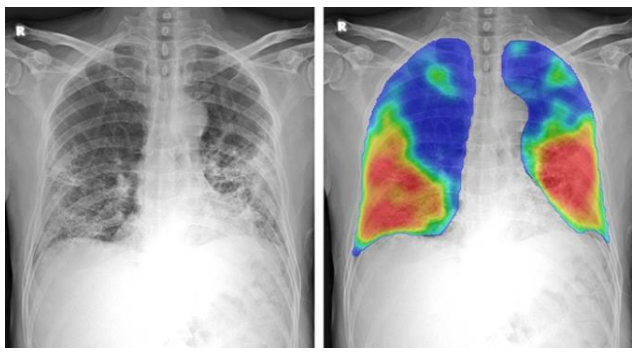

Figure 3. Chest X-ray showing COVID-19 symptoms.

B. Baseline classifier

Before measuring the performance of the proposed method, a baseline solution with k-NN (k-Nearest Neighbor) classifier was developed as a baseline. Measurement results were obtained by comparing and classifying 1-NN (nearest neighbor) characteristic vectors for averaging over 5 random test sets, as shown in the Table 2.

Number of train and test classes	accuracy
4 train – 2 test	0.6980
4 train – 4 test	0.6499
6 train – 2 test	0.6563
6 train – 4 test	0.6499
8 train – 2 test	0.6199
8 train – 4 test	0.5567
Average	**0.6384**

Table 2. Results of the k-NN classifier

The data in the table show the following trends: as the number of known classes increases, the accuracy decreases, and the accuracy of the model also decreases by estimating more and more classes within the same class number group.

C. Test scenarios

After the implementation of DVMN classifier, it was necessary to create a comprehensive plan. Three testing scenarios have been developed that are suitable for testing different types of tasks:

i. New world scenario: it measures the ability to classify new classes after learning (e. g., the recognition of new diseases).
   1. phase: learning distance metrics based on set S_B, where there are N_C classes in set S_B.
   2. phase: selection of a support set (disjoint from set S_B) by selecting one (or a few) images per class from M new classes for the classifier.
   3. phase: prediction for unknown images belonging to one of the M new classes.

ii. Standard scenario: after learning, measures the ability to classify in learned classes on new samples (i.e., on a disjoint test set).
   1. phase is the same as in the first scenario (i/1).
   2. phase: select support set from set S_B.
   3. phase: prediction for unknown images belonging to one of the N_C known classes.

iii. Hybrid scenario: after learning, measures the ability to classify in learned and new classes (i.e., the test set includes both known and unknown classes).
   1. phase is the same as in the first scenario (i/1).
   2. phase: selection of support set by selecting one (or a few) images from class K (known and unknown) for each class.
   3. phase: prediction for unknown images belonging to one of the K classes.

The results of each test plan are presented below, where the tables show the accuracy values (i.e., the ratio of the correct decision to the total classification decision). 1/2/5-shot learning was tested using 1/2/5 samples per class, and the notation in the header of the rows is as follows: C <training classes> / C <test classes> / S <samples per class> / E <number of the epochs>.

The results in Table 3 were measured on the data set described earlier with the baseline classifier (i.e., there was no double-view feature as with DVMN), where the images include recordings from multiple perspectives. As a baseline, we were interested in the results of three scenarios simulating different test circumstances. The measured values in the table clearly show that even the baseline classifier is able to classify with relatively good accuracy in the "Standard" scenario; the results of the "New World" scenario provided an encouraging starting point for recognizing unknown diseases as the main goal of the research, using already known diseases.

Number of train, test classes, shots and epochs	New World scenario	Standard scenario	Hybrid scenario
C4/C2/S2/E1	0.920	0.939	0.766
C4/C2/S2/E5	0.924	0.896	0.846
C4/C2/S2/E10	0.898	0.904	0.825
C4/C4/S2/E1	0.620	0.759	0.800
C4/C4/S2/E5	0.760	0.892	0.823
C4/C4/S2/E10	0.742	0.890	0.805
C6/C2/S2/E1	0.779	0.939	0.750
C6/C2/S2/E5	0.800	0.888	0.776
C6/C2/S2/E10	0.793	0.898	0.770
C6/C4/S2/E1	0.759	0.779	0.699
C6/C4/S2/E5	0.648	0.836	0.693
C6/C4/S2/E10	0.708	0.858	0.673
C8/C2/S2/E1	0.960	0.940	0.600
C8/C2/S2/E5	0.884	0.868	0.726
C8/C2/S2/E10	0.872	0.870	0.713
C8/C4/S2/E1	0.680	0.800	0.766
C8/C4/S2/E5	0.720	0.880	0.746
C8/C4/S2/E10	0.690	0.818	0.726
Average	**0.7808**	**0.8696**	**0.7501**

Table 3. Accuracy results of test scenarios

In the Table 3, the three numerical values below each other belong to a common measurement in such a way that the accuracy values after 1, 5, and 10 epochs were measured. In most cases, after the 5th epoch, the training reached the accuracy value after that the system could no longer learn. In order to avoid overfitting, we used the results after the 5th epoch; and the test results presented in the rest of the paper also include learnings up to the first 5 epochs.

D. Classification of unknown diseases, like COVID-19

In the following, the measurements of the "New World" scenario, which simulates the recognition of new diseases, is the topic that gives the main objective of our research. The tables compare the classifiers:

- MN (Matching Network) for only first, and for only second view (and average accuracy of them),
- k-NN classifier for only first, and for only second view (and average accuracy of them),
- DVMN as our proposed method
- k-NN classifier for two views

- MN (Matching Network) for all data (without distinguishing views)
- k-NN classifier for all data (without distinguishing views)

Looking at the results in Table 4, it can be seen that the DVMN method performs best with an average accuracy of 81.2%, even when using a single sample for the one-shot-learning task.

# train, test class, shots	Average of 2 different views		Double view		Without distinguishing views	
accuracies	***Ave. of 2 MN***	***Ave. of 2 k-NN***	***DVMN***	***2-view k-NN***	***MN***	***k-NN***
C4/C2/S1	**0.8566**	0.6560	0.7766	0.6200	0.8380	0.5833
C4/C4/S1	**0.8683**	0.6333	0.8333	0.6366	0.7640	0.6500
C6/C2/S1	0.8149	0.5933	**0.8200**	0.6600	0.7940	0.6499
C6/C4/S1	0.7599	0.5426	**0.7600**	0.5666	0.7240	0.5400
C8/C2/S1	0.7450	0.6205	0.8333	0.6000	**0.8740**	0.5600
C8/C4/S1	0.7900	0.6220	**0.8466**	0.5200	0.7700	0.6199
Average	0.8057	0.6112	**0.8116**	0.6005	0.7940	0.6005
F_1 scores	***Ave. of 2 MN***	***Ave. of 2 k-NN***	***DVMN***	***2-view k-NN***	***MN***	***k-NN***
C4/C2/S1	**0.795**	0.574	0.762	0.510	0.751	0.480
C4/C4/S1	0.743	0.570	**0.795**	0.593	0.746	0.546
C6/C2/S1	0.704	0.524	**0.776**	0.604	0.717	0.587
C6/C4/S1	0.701	0.492	**0.724**	0.530	0.709	0.519
C8/C2/S1	0.725	0.586	0.797	0.549	**0.800**	0.556
C8/C4/S1	0.745	0.577	**0.810**	0.507	0.766	0.532
Average	0.736	0.554	**0.777**	0.549	0.748	0.537

Table 4. Accuracies and F_1 scores of 1-shot learning at New World scenario

# train, test class, shots	Average of 2 different views		Double view		Without distinguishing views	
accuracies	***Ave. of 2 MN***	***Ave. of 2 k-NN***	***DVMN***	***2-view k-NN***	***MN***	***k-NN***
C4/C2/S2	0.9000	0.6828	0.8633	0.6300	**0.924**	0.6980
C4/C4/S2	**0.9000**	0.6616	0.8434	0.5833	0.760	0.6499
C6/C2/S2	0.8459	0.6425	**0.8566**	0.6166	0.800	0.6563
C6/C4/S2	0.7680	0.6649	**0.7966**	0.6500	0.678	0.6499
C8/C2/S2	0.7739	0.6636	**0.9133**	0.5333	0.884	0.6199
C8/C4/S2	0.7340	0.6499	**0.8666**	0.5833	0.720	0.5567
Average	0.8203	0.6608	**0.8566**	0.5994	0.7943	0.6384
F_1 scores	***Ave. of 2 MN***	***Ave. of 2 k-NN***	***DVMN***	***2-view k-NN***	***MN***	***k-NN***
C4/C2/S2	**0.836**	0.561	0.810	0.547	0.826	0.570
C4/C4/S2	**0.792**	0.526	0.772	0.515	0.731	0.535
C6/C2/S2	0.771	0.576	**0.791**	0.536	0.754	0.514
C6/C4/S2	0.707	0.511	**0.745**	0.510	0.667	0.556
C8/C2/S2	0.735	0.595	**0.843**	0.479	0.814	0.547
C8/C4/S2	0.684	0.540	**0.786**	0.524	0.659	0.486
Average	0.754	0.552	**0.791**	0.519	0.742	0.535

Table 5. Accuracies and F_1 scores of 2-shot learning at New World scenario

It can be read from Table 5 that in the case of increasing the number of samples, in the vast majority of test cases, the DVMN method achieves the best classification performance with an average accuracy of 85.7% in 2-shot learning.

At 5-shot learning, significantly fewer samples are available from the second view than from the first, so a comparison of the results would not have been statistically possible. Although fewer test cases were available compared to the previous two measurements (Tables 4 and 5), the results of Table 6 showed that the performance of the DVMN classifier is the best in this case as well, with an average accuracy of 85.4%.

# train, test class, shots	First view		Double view		Without distinguishing views	
accuracies	***first view of MN***	***first view of k-NN***	***DVMN***	***2-view k-NN***	***MN***	***k-NN***
C4/C2/S5	0.8180	0.6333	**0.8433**	0.6400	0.7780	0.6100
C4/C4/S5	0.7580	0.5800	0.7960	0.5200	**0.8159**	0.6400
C6/C2/S5	0.8320	0.6599	**0.924**	0.4400	0.8539	0.6333
Average	0.8026	0.6244	**0.8544**	0.533	0.8159	0.6277
F_1 scores	***first view of MN***	***first view of k-NN***	***DVMN***	***2-view k-NN***	***MN***	***k-NN***
C4/C2/S5	0.746	0.529	**0.770**	0.568	0.710	0.535
C4/C4/S5	0.697	0.520	0.706	0.475	**0.764**	0.510
C6/C2/S5	0.763	0.547	**0.836**	0.437	0.776	0.493
Average	0.735	0.532	**0.771**	0.493	0.750	0.513

Table 6. Accuracies and F_1 scores of 5-shot learning at New World scenario

VI. CONCLUSION

The few-shot learning problem presented in this paper intends to help the work of those working in medicine by analyzing pathological X-ray recordings, using only very few images. Although there are solutions, if the images come from different views, they will not handle these views well. We proposed an improved method, the so-called Double-View Matching Network (DVMN based on the deep neural network), which solves the few-shot learning problem as well as the different views of the pathological recordings in the images. The main contribution of this paper is the convolutional neural network for feature extraction and handling the multi-view in image representation. Our method was tested in the classification of images showing unknown COVID-19 symptoms in an environment designed for learning a few samples, with prior meta-learning on images of other diseases only. We compared the results with k-NN classifiers, with different variants of the Matching Network method (one variant for only one view and another without distinguishing views). The results show that DVMN reaches the best accuracy on multi-view dataset (better than Matching Network as well) at 1-shot, 2-shot, and 5-shot learning.

ACKNOWLEDGMENT

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Info-communications).

REFERENCES

[1] Cohen, J. P., Morrison, P., Dao, L., Roth, K., Duong, T. Q. and Ghassemi, M. (2020). COVID-19 Image Data Collection: Prospective Predictions Are the Future, arXiv:2006.11988, https://github.com/ieee8023/covid-chestxray-dataset

[2] Collier, M., & Beel, J. (2019). Memory-Augmented Neural Networks for Machine Translation. arXiv preprint arXiv:1909.08314.

[3] Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning (ICML 2017), Volume 70 (pp. 1126-1135). DOI: 10.5555/3305381.3305498

[4] Garcia, V., & Bruna, J. (2017). Few-shot learning with graph neural networks. arXiv preprint arXiv:1711.04043.

[5] Geras, K. J., Wolfson, S., Shen, Y., Wu, N., Kim, S., Kim, E., ... & Cho, K. (2017). High-resolution breast cancer screening with multi-view deep convolutional neural networks. arXiv preprint arXiv:1703.07047.

[6] Goodrich, R. K. (1970). A Riesz representation theorem. Proceedings of the American Mathematical Society, 24(3), 629-636. DOI: 10.1090/S0002-9939-1963-0145334-0

[7] Jain, P. (2018). Metric Learning Tutorial. https://parajain.github.io/metric_learning_tutorial/

[8] Kan M, Shan S. and Chen X., (2016). Multi-view Deep Network for Cross-View Classification, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 4847-4855, DOI: 10.1109/CVPR.2016.524

[9] Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In ICML deep learning workshop (Vol. 2).

[10] Kondor, R. (2003). A Short Introduction to Hilbert Space Methods in Machine Learning. Tutorial at Columbia University.

[11] Li, Y., Yang, M., & Zhang, Z. (2018). A survey of multi-view representation learning. IEEE transactions on knowledge and data engineering, vol. 31, no. 10, pp. 1863-1883, DOI: 10.1109/TKDE.2018.2872063

[12] Metzen, J. H., Kröger, T., Schenk, A., Zidowitz, S., Peitgen, H. O., & Jiang, X. (2009). Matching of anatomical tree structures for registration of medical images. Image and Vision Computing, Volume 27, Issue 7, pp. 923-933. DOI: 10.1016/j.imavis.2008.04.002

[13] Mishra, N., Rohaninejad, M., Chen, X., & Abbeel, P. (2017). Meta-learning with temporal convolutions. arXiv preprint arXiv:1707.03141, v3.

[14] Papp, D., & Szűcs, G. (2017). Balanced active learning method for image classification. Acta Cybernetica, 23(2), 645-658. DOI: 10.14232/actacyb.23.2.2017.13

[15] Papp, D., & Szűcs, G. (2018). Double probability model for open set problem at image classification. Informatica, 29(2), 353-369. DOI: 10.15388/Informatica.2018.171

[16] Ramachandra, B., Jones, M.J., & Vatsavai, R. (2020). Learning a distance function with a Siamese network to localize anomalies in videos. 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), 2587-2596.

[17] Seeland, M., & Mäder, P. (2021). Multi-view classification with convolutional neural networks. Plos ONE, 16(1), e0245230. DOI: 10.1371/journal.pone.0245230

[18] Shyam, P., Gupta, S., & Dukkipati, A. (2017). Attentive recurrent comparators. In Proceedings of the 34th International Conference on Machine Learning - Volume 70 pp. 3173-3181. DOI: 10.5555/3305890.3306009

[19] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[20] Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In Advances in neural information processing systems, pp. 4077-4087. DOI: 10.5555/3294996.3295163

[21] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., & Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1199-1208).

[22] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 1-9. DOI: 10.1109/CVPR.2015.7298594

[23] Szűcs, G., & Henk, Z. (2015). Active clustering based classification for cost effective prediction in few labeled data problem. Academy of Economic Studies. Economy Informatics, vol. 15, no. 1/2015, pp. 5-13.

[24] Szűcs, G., Papp, D., & Lovas, D. (2014). Viewpoints combined classification method in image-based plant identification task. In Working Notes for CLEF 2014 Conference, vol. 1180, pp. 763-770.

[25] Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D. (2016). Matching networks for one-shot learning. Advances in neural information processing systems, 29, 3630-3638. DOI: 10.5555/3157382.3157504

[26] Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys (CSUR), 53(3), 1-34. DOI: 10.1145/3386252

[27] Zhao, A., Balakrishnan, G., Durand, F., Guttag, J. V., & Dalca, A. V. (2019). Data augmentation using learned transformations for one-shot medical image segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 8543-8553. DOI: 10.1109/CVPR.2019.00874

[28] Ziko, I., Dolz, J., Granger, E., & Ayed, I. B. (2020). Laplacian regularized few-shot learning. Proceedings of the 37th International Conference on Machine Learning, PMLR 119, pp. 11660-11670.

Gábor Szűcs has received MSc in electrical engineering and PhD in computer science from the Budapest University of Technology and Economics (BME) in 1994 and in 2002, respectively. He is an at Department of Telecommunications and Media Informatics of BME. His research areas are data science, artificial intelligence, deep learning, content-based image retrieval, multimedia mining. The number of his publications is more than 100. He is the president of the Artificial Intelligence Section of HTE (Scientific Association for Infocommunications), he is the leader of the research group DCLAB (Data Science and Content Technologies). He has earned János Bolyai Research Scholarship of the Hungarian Academy of Science some years ago.

Marcell Németh was born in Budapest, Hungary in 1997. He is a MSc student at the Budapest University of Technology and Economics, Faculty of Electrical Engineering and Informatics as computer engineering. His research activities are data science, data series analytics, image recognition and machine learning. He took second place in the Students' Scientific Conference 2020 at Budapest University of Technology and Economics.