This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

REBUILD

ICT-enabled integration facilitator and life rebuilding guidance

Project start date: 01/01/2019 | Duration: 36 months

Deliverable: D3.1 Users' profile modeling

DUE DATE OF THE DELIVERABLE: 30-09-2019

ACTUAL SUBMISSION DATE: 04-10-2019

Project	REBUILD – ICT-enabled integration facilitator and life rebuilding guidance
Call ID	H2020-SC6-MIGRATION-2018-2019-2020 – DT-MIGRATION-06-2018
Work Package	*WP3 – Data Analysis and skills matching*
Work Package Leader	*Universidad Politécnica De Madrid UPM*
Deliverable Leader	*Centre for Research and Technology Hellas CERTH*
Deliverable coordinator	Theodoros Semertzidis (*CERTH*) – theosem@iti.gr
Deliverable Nature	Report
Dissemination level	Public (PU)
Version	1.0
Revision	Final

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

Document Info

Authors

Author name	Organization	E-Mail
Theodoros Semertzidis	CERTH	theosem@iti.gr
Anastasia-Sotiria Toufa	CERTH	rtoufa@iti.gr

Document History

Version #	Author name	Date	Changes
0.1	Theodoros Semertzidis	18-07-2019	Starting version
0.2	Anastasia-Sotiria Toufa	20-08-2019	State of the art and introduction
0.3	Anastasia-Sotiria Toufa Theodoros Semertzidis	04-09-2019	Methodology and updates on available data
0.4	Anastasia-Sotiria Toufa Theodoros Semertzidis	27-09-2019	Submitted for internal review
1.0	Maria Amata Garito (UNINETTUNO)	04-10-2019	Final submitted version

Document Data

Keywords	*Profile modeling, profile analysis, embeddings*
Editor Address data	Name: Theodoros Semertzidis Partner: CERTH Address: 6th km Charilaou Thermi Rd. Thermi, Thessaloniki, Greece Phone: +30 2310 464160 ext. 142 Email: theosem@iti.gr
Delivery Date	04-10-2019
Peer Review	*Maria Amata Garito (UNINETTUNO)*

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

EXECUTIVE SUMMARY

This deliverable is reporting the work done in REBUILD project's WP3 on the analysis of data to provide profile modeling and further analysis. This part of the work is providing the foundations for creating the appropriate methodologies and functionalities for personalized services and matching of needs and available resources.

The first part of the report is focusing on the academic literature review and the technical approaches that can be followed for profile analysis. Next, a section that is presenting the available open data and public datasets that may be used for modeling and training the system are presented. A detailed presentation on the proposed methodologies is then provided. The details of the provided methodology will be further updated in future deliverables of WP3 when more information for the available input data will be available.

It is important to note that the deliverable is studying only the technical considerations that should be taken into account, but it is tightly following all the legal and ethical national and European rules that come with the new GDPR law.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

Table of Contents

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

INDEX OF FIGURES

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

1 INTRODUCTION

The project REBUILD aims at improving migrants and refugees' inclusion through the provision of a toolbox of ICT-based solutions aimed to enhance both the effectiveness of the services provided by local public administration and organizations, and the life quality of the migrants.

This project follows a user-centered and participated design approach, aiming at addressing properly real target users' needs, ethical and cross-cultural dimensions, and at monitoring and validating the socio-economic impact of the proposed solution. Both target groups (immigrants/refugees and local public services providers) will be part of a continuous design process; users and stakeholders' engagement is a key success factor addressed both in the Consortium composition and in its capacity to engage relevant stakeholders external to the project. Users will be engaged since the beginning of the project through interviews and focus groups; then will be part of the application design, participating in three Co-Creation Workshops organized in the three main piloting countries: Italy, Spain and Greece, chosen for their being the "access gates" to Europe for main immigration routes. Then again, in the 2nd and 3rd years of the project, users' engagement in Test and Piloting events in the three target countries, will help the Consortium fine-tuning the REBUILD ICT toolbox before the end of the project.

The key technology solutions proposed are:

- GDPR-compliant migrants' integration related background information gathering with user consent and anonymization of personal information;
- AI-based profile analysis to enable both personalized support and policy making on migration-related issues;
- AI-based needs matching tool, to match migrant needs and skills with services provided by local authorities in EU countries and labour market needs at local and regional level;
- a Digital Companion for migrants enabling personalized two-way communication using chatbots to provide them smart support for easy access to local services (training, health, employment, welfare, etc.) and assessment of the level of integration and understanding of the new society, while providing to local authorities data-driven, easy to use decision supporting tools for enhancing capacities and effectiveness in service provision.

1.1 SCOPE AND OBJECTIVES OF THE DELIVERABLE

REBUILD project's objective is to provide a toolbox of ICT-based solutions that will help to the smooth integration of refugees and migrants. REBUILD refers both to the facilitation of the local authorities' management procedures and to the migrants' life quality improvement. To achieve these goals, REBUILD is designed as a user-centered application that attempts to recognize users' needs and give them personalized recommendations and targeted solutions. To do so, personal information for each migrant is required in order to learn profile patterns and link to needs and resources. For the gathering of those necessary data, all users will have to consent in order to provide anonymized, GDPR-compliant information that will be used by AI-based methods. More specifically, the proposed technological solutions include an AI-based profile analysis to enable the personalized support, an AI-based matching tool in order the migrants' needs and skills to be matched with services provided by local authorities in each pilot country and a set of tools such as a chatbot or audio visual

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

communication to enable personalized two-way, effective communication between the final users, i.e migrants and local service providers.

This deliverable focuses on the state of the art and a preliminary proposed methodology to build such profiles models that will enable the personalized functionalities of REBUILD platform. The deliverable is reporting the work done in WP3 and specifically, in task 3.1 but also contains synergies with other tasks of the work package.

It should be noted here that this work is answering the technical question of "how to perform profile modeling from raw personal data of different types and with missing values?". There is no discussion of what data will be actually available and how these data will enter the system, which is a discussion that will be answered also within the co-creation workshops that REBUILD will organize in the following months.

The first part of the report is discussing different state of the art approaches that are able to handle such data for creating vector embeddings i.e. vector representations of the raw information, that will be used for the different personalization and matching approaches. Next, a discussion on the available datasets that may be used for training is also performed. Finally, a detailed discussion on the proposed methodology is presented with design decisions and explanatory figures.

Since, REBUILD is a user-driven project the information of this deliverable will be updated with the decisions taken within the co-creation and the other REBUILD workshops with migrants and stakeholders to better approach their actual needs.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

2 ACADEMIC LITERATURE REVIEW AND RELATED APPROACHES

2.1 INTRODUCTION

Machine learning is a subset of artificial intelligence which exploits deep learning techniques in order to create models that have the ability to learn using previous experience and knowledge just like humans do. In recent years, artificial intelligence, machine learning and deep learning are terms that have become almost equivalent. The two main categories of machine learning is supervised and unsupervised learning. In the former the algorithm uses an extra information that represents the desired output and works as a supervisory signal while in the latter, this information is not available and the algorithm discovers unknown patterns and correlations among the data in order to model their distribution. In the next subcategories supervised and unsupervised learning are presented, providing a detailed discussion about the main tasks being solved while new approaches and methods are analyzed.

2.1.1 SUPERVISED LEARNING

The most representative task for supervised learning is classification of objects in a set of classes. In classification tasks, each input data is associated with a corresponding label that represents the class information and it is working as the supervisory signal. The model used is trained on a known dataset in order to learn a function that maps each input sample to the corresponding output. The goal of the model is to be able to work just as well in new unseen samples. Trivial applications for classification regarding the number of classes is binary classification in which there are only two classes and multiclass classification where there are more than three classes. In both cases the input sample belongs to only one class. The first approaches of multiclass classification treated the problem as transformation or extension to the binary case but with the high performance of deep learning models works such as (He, Zhang, Ren, & Sun, 2016; G. Huang, Liu, van der Maaten, & Weinberger, 2017; Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016; Simonyan, & Zisserman, 2014) outperformed previous attempts.

A challenging task is multilabel classification where multiple labels have been assigned to one object. An ineffective tactic is to treat multi-label classification as a set of binary classification tasks one for each label. This approaches ignore the overall topology structure of the data. A novel work proposed in (Wang, Jiang, Yang, Mao, Huang, Huang, & Xu, 2016) transform the labels into embedded vectors, in order to employ the correlations and the semantic dependencies between them. In (Gong, Jia, Leung, Toshev, & Ioffe, 2013) a deep convolutional network is used with a ranking objectives for the training procedure. Another line of work uses attention mechanisms, such as the method proposed in (Zhu, Li, Ouyang, Yu, & Wang, 2017) where a spatial regularization network capture both semantic and spatial relations and the work in (Z. Wang, Chen, Li, Xu, & Lin, 2017) where a spatial transformer layer and LSTM units are used to capture the label correlation. A recent work proposed in (Chen, Wei, Wang, & Guo, 2019) uses a Graph Convolutional Network (GCN) to build a directed graph over the object labels and the model learns to map the label graph into a set of inter-dependent object classifiers. Multi-label classification models trained on text data has been proposed in (Nam, Kim, Loza Mencía, Gurevych, & Fürnkranz, 2014) while in (Liu, Chang, Wu, & Yang, 2017) a method for handling an extremely large collection of labels is proposed.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

Traditional deep learning models have shown great performance in classification tasks with the only requirement to be trained on an understandably large dataset. In contrast, humans can recognize objects they have seen only a few times and they can learn new concepts fast by combining their previous experiences and knowledge they already have. Because of those characteristics, they are considered to have versatility, a key aspect of intelligence. A different, new perspective in machine learning but at the same time closer to the human way of thinking is known as meta-learning. The objective of meta-learning is to create models capable to learn the learning process and when this is achieved their previous experience can be applied to new unseen tasks. For supervised learning the most representative task is few-shot classification for images, in which the model learns to classify an image among some classes, given just a few examples per class. For the evaluation, the model is tested on unseen classes. The extreme case of few-shot learning is zero-shot learning in which the model tries to classify a sample in a class for which no training sample has been seen. In Section 2.3 an extensive analysis of the recently proposed few-shot classification methods is performed, and indirectly the main approaches of meta-learning methods are described.

2.1.2 Unsupervised Learning

The other aspect of learning is unsupervised learning with the most representative task to be clustering. The goal of clustering is to find a partition of the initial data points in order to keep similar points in the same cluster, while dissimilar ones in different clusters. But clustering is an ambiguous task where questions like what constitutes a cluster, what is the proper metric to measure similarity between data points and how the final clustering can be validated has to be clarified. Because of those issues which can not be clearly defined there have been proposed a lot of different clustering methods.

One approach is connectivity-based algorithms known as hierarchical. (Corpet, 1988; Ding & Xiaofeng He, 2002; Johnson, 1967; Karypis, Eui-Hong Han, & Kumar, 1999). These algorithms produce an extensive hierarchy of clusters instead of a partition of the data points. They form clusters by connecting objects based on their distance and as the algorithm progresses the already created clusters are merged with each other. Depending on the strategy they use for the merging of the clusters they can be separated in agglomerative (bottom-up) and divisive (top-down) type. Apart from the distance function that has to be defined, a linkage criterion is also necessary and the most common choices are single-link, complete-link, and average-link distance. A different approach that also uses a linkage criterion is based on density algorithms in which clusters are formed in areas with higher density than others (Cao, Estert, Qian, & Zhou, 2006; Kriegel, Kröger, Sander, & Zimek, 2011). Data points in the sparse area are considered as noise or border points. The most popular density based algorithm for clustering is DBSCAN proposed in (Ester, Kriegel, Sander, & Xu, 1996) and since then many variations have been proposed (Ali, Asghar, & Naseer Ahmed Sajid, 2010; Sander, Ester, Kriegel, & Xu, 2010).

For many years centroid-based algorithms were the first choice for clustering with k-means proposed in (MacQueen, 1967) being the most representative. These algorithms use a central vector to represent the center of the cluster that it may not be necessarily a member point of the dataset, and each point of the dataset is assigned to the nearest cluster center. After the point's assignment, new central vectors are computed and the process continues in an iterative way. An important limitation is that the number of clusters have to be predefined and clusters should have similar size. There are many variations in the literature about how central vectors are computed, how to initialize them and how the number of clusters is estimated.

In many works the representation of the data points is considered of vital importance in order to achieve good clustering results. To this end clustering is considered and treated as two independent procedures. A common technique is to perform a dimensionality reduction method to initial data points as a first step, and then to apply

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

the clustering algorithm itself. The key idea behind this is that the new embedded space can be more representative regarding the original distribution of the data and clustering could be applied more efficiently. Spectral clustering proposed in (Ng, Jordan, & Weiss, 2001; von Luxburg, 2007) is a common clustering method with great performance and it is working in a similar way. It uses the spectrum i.e. the eigenvalues of a similarity matrix to perform dimensionality reduction before clustering. The goal of this nonlinear dimensionality reduction is to maintain the geometrical structure of the initial data points and so, points that are close to initial data space, can be also close to the embedded space. Motivated by the success of deep learning, it was inevitable for deep neural networks (DNN) not to be used for clustering. In Section 2.2 the recent deep learning methods for clustering are presented and in Subsection 2.2.1 novel methods for spectral clustering with deep learning techniques are analyzed.

2.2 Clustering With Deep Learning Methods

Recently, DNN models have been used for clustering, but those methods keep also separate the dimensionality reduction and clustering as two different procedures. The works in (Andrew, Arora, Bilmes, & Livescu, 2013; Vincent, Larochelle, Lajoie, Bengio, & Manzagol, 2010; Ng, 2011) use the proposed models as a preprocessing procedure before clustering, that is designed independently from the subsequent clustering stage. However, since no clustering objective is incorporated in the learning process, the embedded output may be not suitable for the clustering this is why a new perception in deep clustering methods came up. Those methods show great performance combining embedding and clustering in a joint procedure. The proposed methods focus on learning a feature space in which clustering is expected to have better results but at the same time the clustering performance is included in the optimization. With this approach the DNN model can capture the distribution of non linear input data more effectively and at the same time refine its performance taking into consideration the current results of the clustering. Thus, the new representation of the input data will be more suitable for clustering.
One of the first works in that direction with great success is DEC model proposed in (Xie, Girshick, & Farhadi, 2015). The model learns a parameterized non-linear mapping from the data space X to a new lower-dimensional feature space Z and a clustering objective is optimized jointly using stochastic gradient descent. Because the method is applied in unlabeled data there is not a supervisory signal, thus the authors proposed the minimization of Kullback–Leibler (KL) divergence in an auxiliary target distribution derived by a soft cluster assignment to refine the clustering. Specifically, given a first estimation of the mapping f_θ and the initial cluster centroids, the optimization step is performed in two steps. In the first step, a soft assignment between the embedded points and the cluster centroids is computed and in the second step the f_θ mapping is updated by taking into account the assignments with high confidence of the auxiliary target distribution. As for the deep neural network a stacked autoencoder is used and the initialization of its parameters is done layer by layer, with each layer being a denoising autoencoder trained to reconstruct the previous layer's output after random corruption as proposed in (Vincent et al., 2010).

A similar work is proposed in (Yang, Fu, Sidiropoulos, & Hong, 2016) which makes the same assumption about the non-linear mapping from input data to a new representation that is more convenient for the clustering. The authors proposed Deep Clustering Network (DCN), a deep autoencoder network which parameters are initialized with the method used in (Bengio, Lamblin, Popovici, & Larochelle, 2006). The optimization procedure includes the embedding to an optimal latent space in which k-means algorithm can be effectively used. Problems with the combination of model's nonlinearity and integer constraints k-means algorithm produce, are faced with an

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

alternating stochastic gradient method by separating the entire optimization problem into subproblems keeping every time a different component fixed.

In (Shah & Koltun, 2017) the optimization of a continuous global objective based on robust statistics allows heavily mixed clusters to be untangled. The clustering and dimensionality reduction tasks are jointly solved and the algorithm can be applied efficiently to high dimensions and large datasets. Based on a previous work, a deep approach is proposed in (Shah & Koltun, 2018) in which dimensionality reduction is performed by a deep autoencoder network and clustering is performed in the embedded space. Noteworthy point in this work is that the clustering procedure does not require the number of clusters to be set in advance and there is no need of reconfiguration of the objective like discrete reassignment of data points to centroids or merging the clusters in agglomerative procedures. A recent work with high performance presented in (Dizaji, Herandi, Deng, Cai, & Huang, 2017) proposes DEPICT model, a model consists of two parts. The first part is a deep convolutional autoencoder with a multinomial logistic regression function i.e. a softmax layer on top of it. This softmax layer works as a discriminative clustering model which is trained on the relative entropy minimization. The model's performance is very high and this is accomplished by the regularization term which penalizes the creation of unbalanced clusters and prevents allocating clusters to outlier samples.

A different approach has been proposed in (Yang, Parikh, & Batra, 2016) which uses a Convolutional Neural Network (CNN) to represent the data and a recurrent framework to express the clustering procedure. The method uses agglomerative clustering (Chidananda Gowda & Krishna, 1978) as a more reliable process which can be represented as a recurrent framework also. More specifically, reliability is transfused because of the over-cluster created in the beginning of the training when the representation of the data is not efficient yet. A partial unrolling strategy is used, in which a number of clusters is merged in every timestep and the CNN parameters are updated for a fixed number of iterations. Thus, the whole clustering procedure can be interpreted as a recurrent process.

2.2.1 Spectral Clustering

Spectral graph theory is a field in mathematics which studies the properties of a graph. It includes nonlinear techniques that are used in order to find a new embedding of data in a lower dimensional space which preserve the geometrical structure of the initial data. With this kind of transformation clustering algorithms can be applied with better results. One of those techniques and one of the most successful clustering algorithms is spectral clustering (Jianbo Shi & Malik, 2000; Ng et al., 2001; von Luxburg, 2007).The method considers data points as nodes of a graph and eigendecomposition is applied in the Laplacian matrix of the graph also called affinity matrix in spectral clustering scope. There are many different ways to construct the Laplacian matrix, most of them by computing the pairwise similarities between data points. For the affinity matrix W a common method for its creation is the Euclidean pairwise distance of data points which is a combination of nearest neighbor algorithm and a Gaussian kernel with scale σ. The values of W are given by the expression:

$$W_{ij} = exp\left(\frac{-\| x_i - x_j \|^2}{2\sigma^2}\right) \quad (1)$$

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

The expression is applied if sample x_j is among the nearest neighbors of x_i, otherwise their distance is considered zero, $W_{ij} = 0$. The new embedded data are the eigenvectors arising from the eigenanalysis and for the final clustering, k-means algorithm is applied in the embedded data.

There are many advantages in spectral clustering:

1. It does not make strong assumptions about the structure of the data and therefore it can handle efficiently non-convex clusters. In contrast with k-means algorithm which assumes spherical clusters with similar variance and density.
2. Optimization is applied by minimizing the pairwise distance between points and the optimal result can be found analytically
3. It outperforms other clustering algorithms such as k-means.

The disadvantages of the method is the use of k-means algorithm in the final step, which provides a non deterministic result and consequently all the disadvantages that accompany k-means. The construction of the proper affinity matrix is quite difficult because the affinity graph can influence the clustering results significantly. The main difficulty in its construction is choosing a proper similarity metric that has to be representative of the data structure. For spectral clustering the most important deterrent is the computational cost of Laplacian matrix eigenvectors, which makes prohibitive the apply of spectral clustering in large datasets. Another shortcoming is that it can't be applied in new, unseen data points since the calculation of the new embedded vectors takes into account the whole dataset.

A novel deep learning clustering approach for discriminative embedding and spectral clustering is proposed in (X. Yang, Deng, Zheng, Yan, & Liu, 2019). It exploits the ability of autoencoder network to capture high dimensional probability distributions of the input data without supervised information. In order to obtain the latent representation where input data can be better separated for clustering, usually the reconstruction loss is used in the optimization procedure. In contrast, the authors support that the discriminative ability of the latent representation has no substantial connection with the reconstruction loss. They propose a reconstruction constraint for the latent representations and their noisy versions to make it more robust to noise. Then the mutual information estimation is utilized to provide more discriminative information from the inputs. The decoder can be considered as a discriminator that decides if the latent representations are representative. In the final step the latent representation is embedded in the eigenspace through spectral clustering and the model provides the final optimal clusters. The overall architecture is presented in Figure 1.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

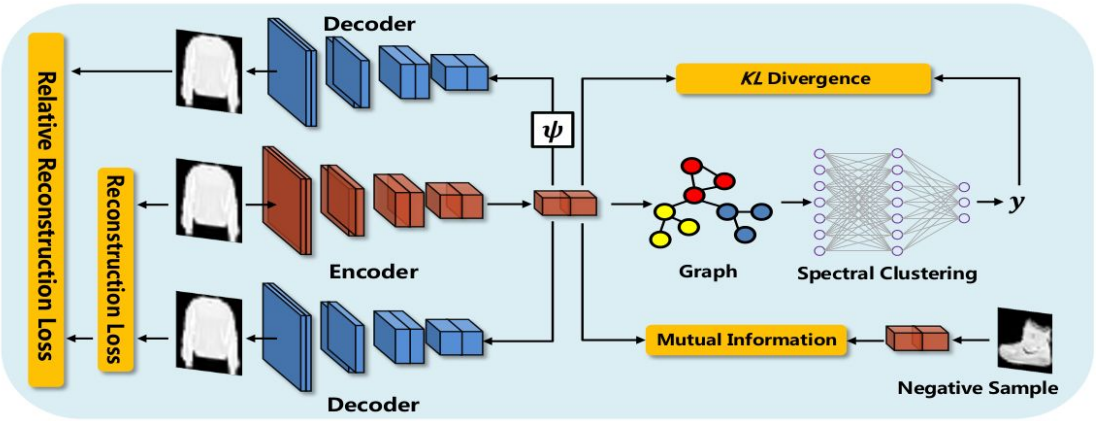

Figure 1:The overall architecture of Dual Autoencoder Network, (image from: X. Yang et al., n.d.)

In (Tzoreff, Kogan, & Choukroun, 2018) the proposed model is optimized with respect to a discriminative pairwise loss function in which the encoder provides an approximation of the maximum likelihood to estimate the distribution diversity between the latent representation and the inputs. SANet model proposed in (Wang, Hilton, & Jiang, 2019) is a spectral analysis network for unsupervised deep representation learning and it is based on multiple consecutive spectral analysis procedures for image clustering tasks. The model performs spectral clustering analysis into multiple layers. At first, it uses patch level information to compute the local similarity of an image and in the next layers it combines the previous results to perform extra spectral analysis. The network provides the embedded vectors of initial input data in a representation that is more suitable for clustering.

SpectralNet model proposed in (Shaham et al., 2018) tries to handle the disadvantages of spectral clustering introducing a novel deep learning approach which uses constrained stochastic optimization for the training of the model. The constraint that is used has to do with the orthogonality of the model's output and it is enforced with a linear layer whose weights are set by the QR decomposition of its inputs. After the training of SpectralNet a function that maps the input data points to their associated spectral representation is provided and each data point is accompanied with a cluster assignment.
Some recent works combine spectral clustering and multi-view data. Multi-view data is a new definition of data that have been collected with different measuring methods, they refer to different properties but all of them are related to one specific object. This kind of data are used even more in real world applications in various domains such as biological, social, computer or sensor network applications (Zhao, Xie, Xu, & Sun, 2017) Essentially, multiview data can be considered as a data structure that uses different views to represent heterogeneous features of the same object. Every different category of those features represent a single-view of the object. For example iris, fingerprint, and signature are different aspects of a person's identity. All those features work complementary thus they subserve with the recognition of a person. Multi-view data are used in order to receive more accurate and reliable results by exploiting all the different features they can provide.
MultiSpectralNet proposed in (Huang, Ota, Dong, & Li, 2019) is a deep learning approach for spectral multi-view clustering and it is based on SpectralNet. Just like SpectralNet, MultiSpectralNet learns a mapping of the input data points in the spectral eigenspace of their associated graph Laplacian matrix. The model can handle multi-view data and provides a function that maps new unseen data points to its spectral embedding coordinates. Because of the use of constrained stochastic optimization, the model can handle large-scale datasets too.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

As shown in Figure 2, at first each view of data is processed independently and the fusion of the results is taking place in the final stage. The affinity matrix for each one of the u views is provided by a Siamese network. In the embedding stage the model trains u independent branches in parallel in order to find the approximate low-dimensional embedded eigenspace that satisfies the orthogonalization of the outputs. In the last stage, the previous single-view embedded eigenspaces providing complementary information about the data, are fused to obtain the final eigenspace of the multiview data points. After the training of the model, k-means algorithm is applied to the embedded data points to obtain the result of the clustering.

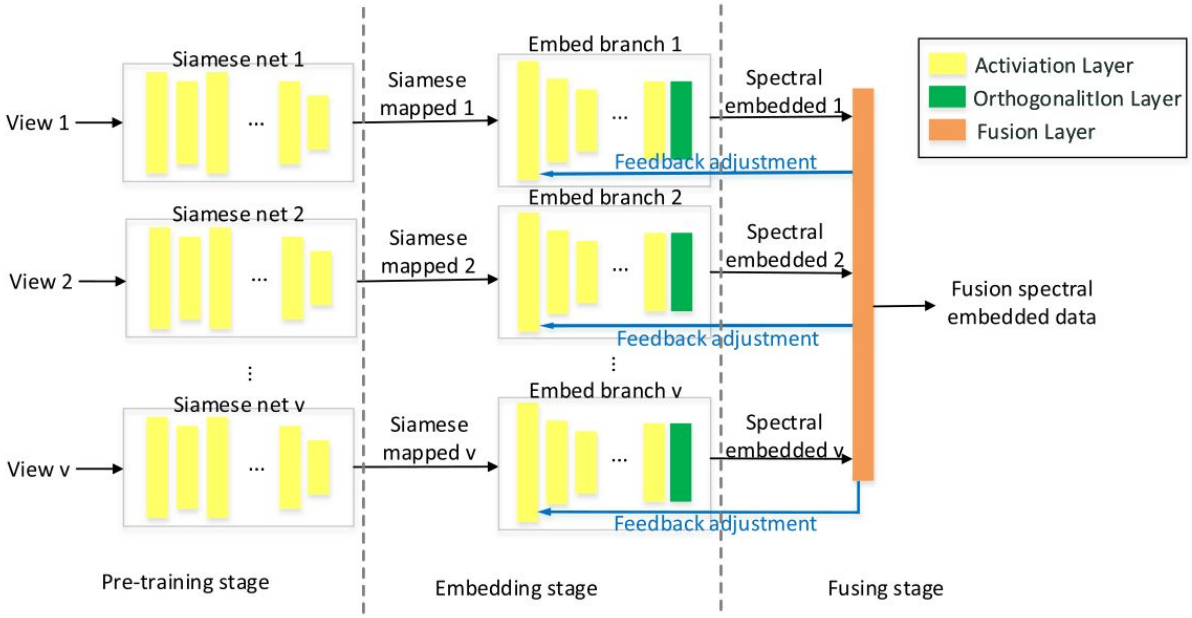

Figure 2: The architecture of MultiSpectralNet, (image from: S. Huang et al., 2019)

A similar problem is solved by the work proposed in (Gheche, Chierchia, & Frossard, 2018), OrthoNet is a deep learning model that tries to handle clustering of network nodes which contain additional information about the edges of the network. The additional information is represented in a multi-view data structure. The proposed model is a two-step algorithm in which the geometric mean of Laplacian matrices associated to each layer of the network is computed. The aggregation of those matrices leads to a graph representation with the form of a symmetric positive definite matrix which takes into account the topology of the network. In the second step a neural network is used for computing the node embeddings and instead of an orthogonality constraint, a differentiable cost function is optimized via gradient descent. The advantage of this extension is that during the training of the model there is no alternation between orthogonalization and gradient step such as in SpectralNet.

2.3 Few-shot Classification

In few-shot classification, a deep learning model classifies samples in their corresponding class while it has been trained on just a few training samples per class. As mentioned before, a common way to solve this task are methods based on meta-learning. Those methods propose to train a model on a large number of different tasks and then to test its ability to be applied in new unseen tasks, thus the model's optimization is based on a

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

distribution of known and unknown tasks . Each of these tasks is associated with a different dataset D that contains samples with their corresponding labels.

There are two different views of meta-learning. In the first one, training and testing phases can be assumed as the same procedure. Having in mind the traditional learning process in which many data points are used for the model's training, and in conjunction with the previous note that every task is associated with a different dataset the extension in meta-learning is that every data point can be assumed as a whole dataset. It is worth noting that for each dataset a whole training procedure is performed. More specifically, the dataset D is split into two parts, the support set S for training and the query set B for testing. Depending on the exact problem to be solved, the task is called *k-shot N-way classification problem* where *N* is the number of classes and *k* is the number of labeled samples per class that are included in the support set S.
In the training phase of meta-learning the model is trained on many different tasks/datasets and in the test phase it is tested in completely new task/dataset. An illustration of the procedure is shown in Figure 3 and the steps that are performed for each task i of the training procedure are:

1. Sample N labels from label set L and make $L_i \subset L$ set.
2. Sample k samples for each one of the N classes and make the support set S_i.
3. Sample k samples for each one of the N classes and make the query set B_i
4. Perform the optimization of the model by using the query set B_i to compute the loss and the backpropagation algorithm for updating the model.

The second view of meta-learning was first proposed in (Ravi & Larochelle, 2016) and provides two modules for the training, the learner and the meta-learner. The learner, can be considered as a low-level network, in essence it is a simple classifier f_θ trained on performing a specific classification task. The meta-learner is a high level model g_φ which is updating the weights of the low-level network (learner). There are two nested processes: the meta-training process of the meta-learner in which the (meta-)forward pass includes several training steps of the learner which includes separate training procedures. In the case of meta-learner the meta-loss measures how well the model is performing in its task i.e training the learner. The loss of the learner can be computed normally as in every training procedure while the loss of meta-learner requires second-order derivatives and it can be computed or at the end of the training procedure or as a summation of individual losses. In meta-learning and, by extension, in few-shot learning there are three different approaches, the model-based, the optimization-based and the metric-based methods which are presented in the next subsections.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

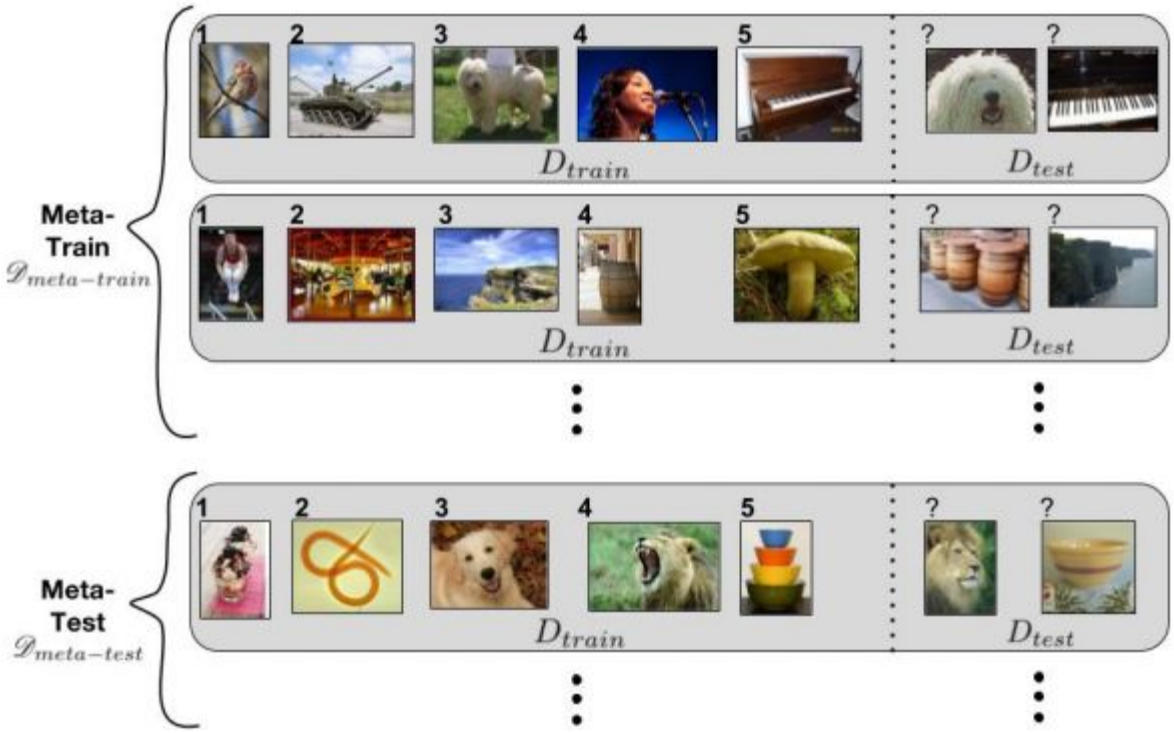

Figure 3: Example of meta-learning set-up for few-shot image classification. In every meta-train set a separate classification task is performed with different training and testing datasets. In this example a 1-shot, 5-class task is performed. The meta-test dataset includes classes not present during the training, (image from: Ravi & Larochelle, 2016)

2.3.1 MODEL-BASED METHODS

The model based methods does not try to find the distribution of the data among different classes, instead they depend on a well designed model created for fast training. The model updates its parameters rapidly within few training steps. This is achieved by its internal architecture or they are computed by another meta-learning model. A common strategy is LSTM based models which use external memory to facilitate the learning process such as Neural Turing Machines (Graves, Wayne, & Danihelka, 2014) and Memory Networks (Weston, Chopra, & Bordes, 2014). The meta-learner uses gradient descent, whereas the learner simply rolls out the recurrent network. Santoro in (Santoro, Bartunov, Botvinick, Wierstra, & Lillicrap, 2016) following the method in (Hochreiter, Younger, & Conwell, 2001) uses a Neural Turing Machine as baseline model and proposes MANN, a model whose memory can encode and capture information of new tasks fast and in the meantime any stored representation is easily and stably accessible. In each training episode the input sample x_t is coming with the label of the previous sample, y_{t-1}, so there is one step offset between the sample and its true label. In this way the memory is forced to hold information for longer and the model has to retrieve the correct label at the right time. More details for MANN model are presented in Figure 4.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

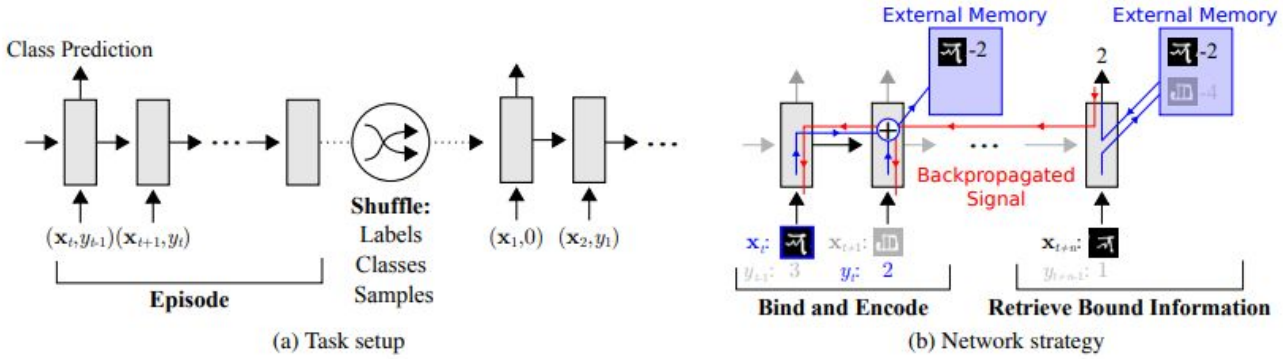

Figure 4: (a) input samples x_t are presented with time-offset labels y_{t-1} to prevent the network from simply mapping the class labels to the output. (b) An external memory is used to store class label information as bound information, which can then be retrieved at a later point for successful classification when a sample from an already-seen class is presented, (image from: Santoro et al., 2016)

A similar work for rapid generalization on new concepts is presented in (Munkhdalai & Yu, 2017) in which the MetaNet model learns a meta-level knowledge across tasks and shifts its inductive biases via fast parameterization for rapid generalization. This is achieved by "fast weights" method (Hinton & Plaut, 1987) in which one neural network is used to predict the weights of another network. The loss gradients in MetaNet are used as meta information to populate models that learn fast weights. MetaNet consists of two learning components, as shown in Figure 5, the learner and the meta-learner and an external memory component that helps to the overall procedure. The meta learner is responsible for fast weight generation by operating across tasks while the base learner performs within each task by capturing the task objective.

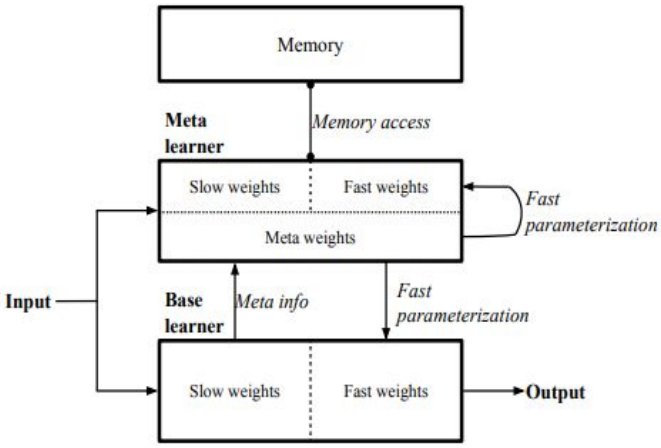

Figure 5: The architecture of MetaNet, (image from: Munkhdalai & Yu, 2017)

In (Gidaris & Komodakis, 2018) a visual system is proposed that can learn novel classes in test time while it does not forget the initial categories on which it was trained on. This is achieved by directly predicting the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

weights of the classifiers for novel classes using an attention-based mechanism that computes the cosine similarity between feature representations and classification weight vectors. The cosine similarity is also used by Qi (Qi, Brown, & Lowe, 2017). In this work the learner is trained with abundant training samples and then it is exposed to previously unseen classes with few samples for each class. The model uses imprinted weights method in which the advantages of convolutional classifiers and embedding methods are combined by using the penultimate layer of a convolutional classifier as an embedding vector. In the imprinted method, the embedded vector provided by a deep convolutional network is copied and is used as a weight in a new network. The procedure of imprinting method is presented in Figure 6.

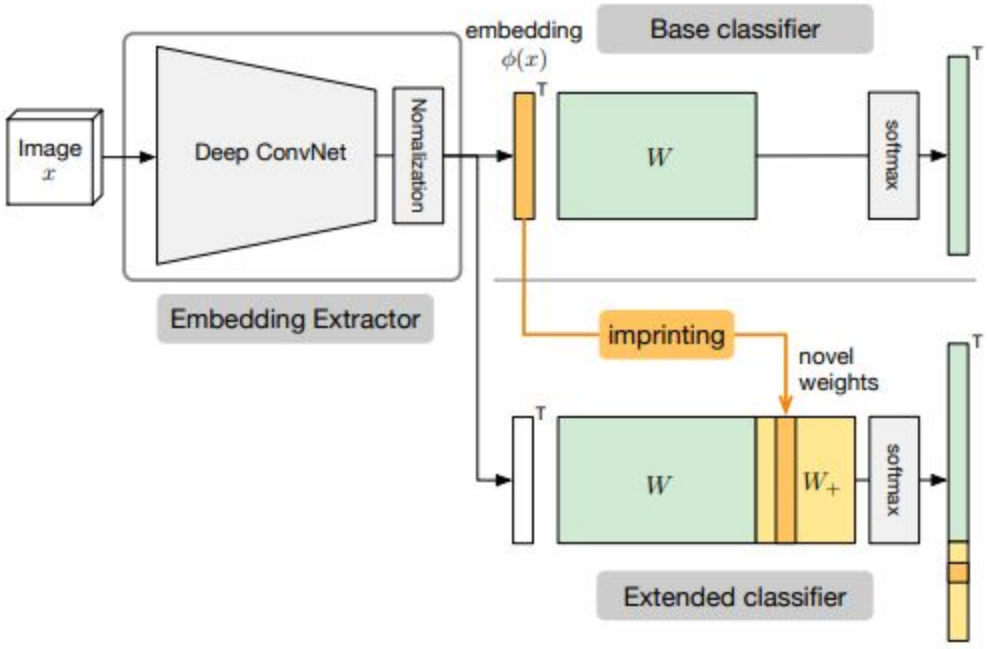

Figure 6: After a base classifier is trained, the embedding vectors of new low-shot examples are used to imprint weights for new classes in the extended classifier, (image from: Qi et al., 2017)

2.3.2 Optimization/Initialization-based methods

Deep learning models are optimized using gradient-based methods via backpropagation. This algorithm is designed to handle a large amount of data and it converge by making a lot of iterations. Those characteristics are completely in contrast with basic principles of few-shot learning, so in order to overcome those aggravating factors in (Hochreiter et al., 2001) were proposed meta-learning models that include a meta-learner who updates the weights of the learner.

The most representative work in this category is proposed in (Ravi & Larochelle, 2016) in which a meta-learner model, based on LSTM units, is learning the optimization algorithm used to train the learner. The learner is a simple classifier which is applied for few-shot learning. LSTM were chosen because of the similarity they present among gradient descent updates in backpropagation and the cell-state update in LSTM. An illustration of the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

computational graph is shown in Figure 7. Another factor that works supplementary is that knowing the history of gradients the gradient update can benefit the optimization procedure. The use of recurrent model is also applied in (Wichrowska et al., 2017) which uses a novel hierarchical RNN architecture, with minimal per-parameter overhead, augmented with additional architectural features that mirror the known structure of optimization tasks.

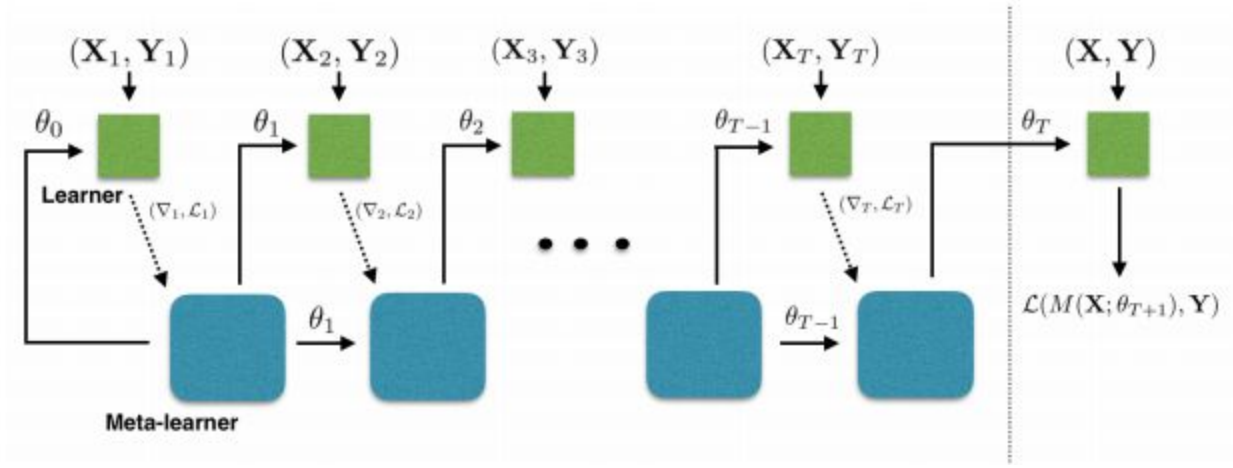

Figure 7: The forward pass of the meta-learner. The dashed lines separates the examples from training and test set. In each episode the learner takes as input a batch from the training set. The LSTM-based meta-learner receives $(\nabla_{\theta_{t-1}} L_t, L_t)$ from the learner and proposes a new set of parameters θ_t for T steps. During the training of the meta-learner the weights do not backpropagate beyond the dashed line, (image from:Ravi & Larochelle, 2016)

A novel work that examines the initialization of a network proposed by Finn in (Finn, Abbeel, & Levine, 2017). In this work a general optimization algorithm is proposed which is compatible with any model that learns through gradient descent. The meta-learner tries to find an initialization that is useful for adapting to various problems but also the adoption should be performed in a small number of steps using only a few examples. Thus the algorithm's aim is to find a representative set of parameters θ. As it is shown in Figure 8 the MAML algorithm optimize a set of parameters θ such that when a gradient step is taken with respect to a particular task i, the parameters are close to the optimal parameters θ_i^* for task i. The advantages of the MAML algorithm is that it does not make any assumption on the form of the model, there are no additional parameters for meta-learning and the learner uses the known gradient descent process.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

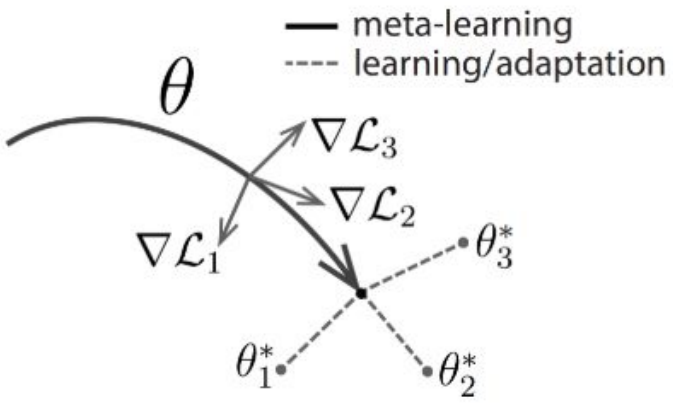

Figure 8: Diagram of the MAML approach which optimizes for a representation θ that can quickly adapt to new tasks, (image from: Finn et al., 2017)

A variation of the model called FOMAML can perform a large number of gradient descent steps ignoring the second derivative terms and provide a good approximation of the original MAML model by reducing the computation cost. The FOMAML model works by remembering the last gradient and applying it to the initial parameters. An extension on that sight is the Reptile algorithm proposed in (Nichol, Achiam, & Schulman, 2018), it works on problems that second derivatives can not be applied. It is a meta-optimization, model-agnostic algorithm that learns an initialization of a model's parameters and provide fast adaptation to new tasks at test time. With Reptile algorithm, a task is sampled, the model is trained on this specific task by multiple gradient descent steps and the initialization of the model's weights move towards the optimal weights on that task. While FOMAML use the last inner gradient update for the model's initialization, Reptile updates the parameters by moving alternately towards two optimal solution manifolds and it converge to the point it minimizes the average squared distance. A visual representation of the process is shown in Figure 9.

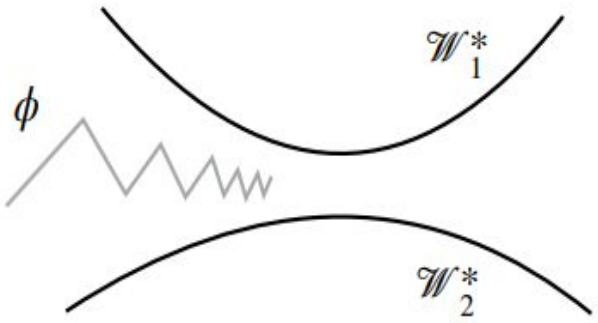

Figure 9: The sequence of iterations of reptile algorithm until its convergence, (image from: Nichol et al., 2018)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

2.3.3 METRIC-BASED METHODS

Metric-based methods are closely related to distance-metric learning and its aim is to learn a metric or similarity/distance function over objects. By learning this function, the model is able to classify unknown objects to the classes they belong or assign them to the right cluster. For one shot classification, for data points x_i of the support set and the corresponding known labels y_i the predicted probability for the final assignment is given by the equation:

$$P_\theta(y|x,S) = \sum_{(x_i,y_i)\in S} k_\theta(x,x_i)\,y_i \quad (2)$$

The factor k_θ is provided by a kernel function that measures the similarity of two data points. Note that if $k_\theta = \frac{1}{k}$ for the k closest samples of x, and 0 for the rest sampes, we obtain the k-nearest-neighbours algorithm. The choice of kernel function is a determining factor for the successful training of the meta-learner which can eventually learn a metric or distance function. It should be mentioned that the metric is problem dependent and and it should represent efficiently the relationships between input data and the corresponding tasks.

A representative model for metric learning is Siamese networks. Siamese networks were first proposed for verification tasks in (Bromley et al., 1993) but in recent years it is a popular choice for metric learning methods. It is about a network composed of two identical sub-networks sharing the same weights and parameters. The sub-networks' output is joined and trained in order to learn the similarity between the input pair. During the training, feature extraction is performed for both inputs and in the joined representation the distance between inputs is computed. In evaluation phase an unknown input is compared with an already known sample and depending on the result, the unknown input is considered as similar or dissimilar. The first version of the model uses Time Delay Neural Network (TDNN) (Waibel, Hanazawa, Hinton, Shikano, & Lang, 1989) while the current approaches use Deep Convolutional Networks as shown in Figure 10 exploiting their better performance. Siamese networks can be used as an embedding network that transforms input data in a new embedding space where data can be effectively separated into the associated classes.

This idea is applied in one-shot learning in the work (Koch, 2015). In training phase the Siamese network learns if two images belong to the same class. This is achieved by computing the L_1 distance between the embedded vectors $f_\theta(x_i), f_\theta(x_j)$ which resulted from the convolutional sub-networks, and then a fully connected layer with sigmoid activation function is applied for the final class predictions. In the test phase the model's aim is to verify if new examples match with a template among known classes. Specifically, it compares all the image pairs formed by the test image and every image in the support set, and among all the comparisons this one with the highest probability give the final predicted class.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

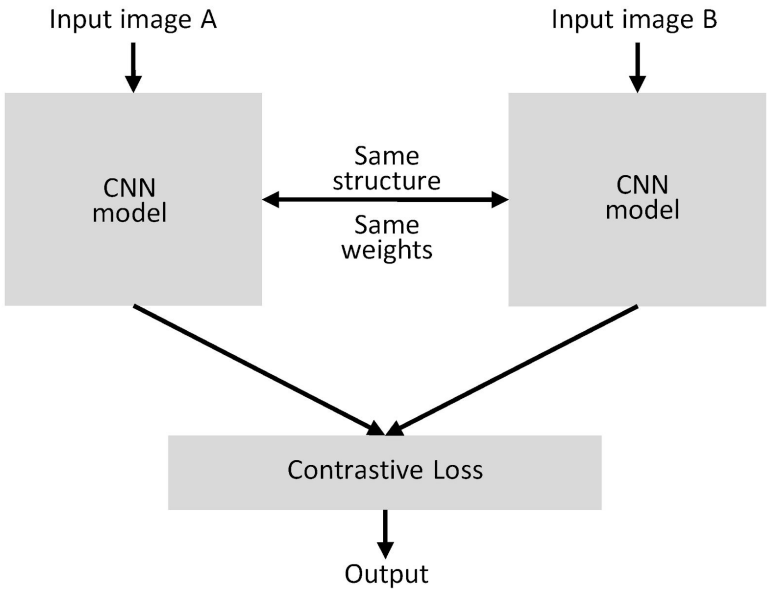

Figure 10: The architecture of Siamese Network, (image from:Jeong, Lee, Park, & Park, 2018)

Matching Networks proposed in (Vinyals, Blundell, Lillicrap, Kavukcuoglu, & Wierstra, 2016), is the first attempt in providing an end-to-end classifier that combines the embedding and classification procedures and it is both trained and tested on the same task. As a representative method for metric-based few-shot learning methods, Matching Networks compute the output class by equation (2). As regards the kernel function k_θ a differentiable attention kernel $a(x, x_i)$ is used, given by the equation:

$$a(x, x_i) = \frac{exp(cosine(f(x), g(x_i)))}{\sum_{j=1}^{k} exp(cosine(f(x), g(x_i)))} \quad (3)$$

where f, g functions are two embedding functions for the query samples and the support set respectively. Both functions use convolutional neural networks providing a completely differentiable model that can be trained end to end with stochastic gradient descent. An overview of the proposed architecture is shown in Figure 11.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

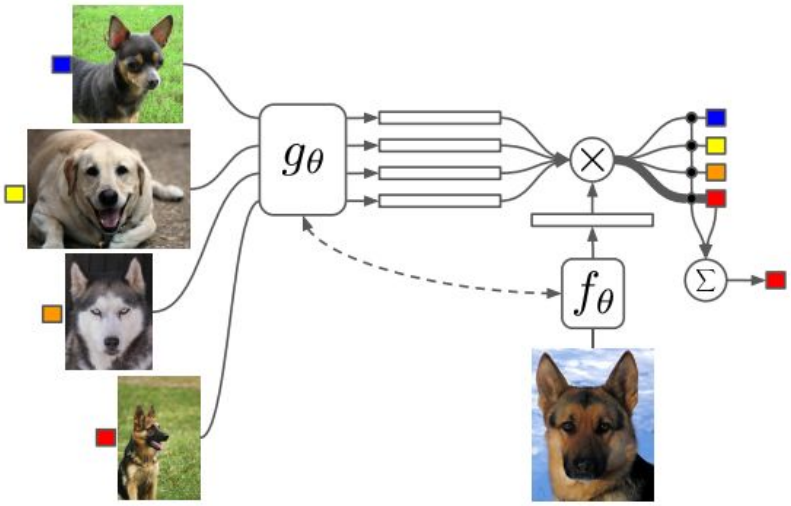

Figure 11: Matching Networks architecture, (image from: Vinyals et al., 2016)

Prototypical Networks (Snell, Swersky, & Zemel, 2017) is a recently proposed model with high performance, extends Matching Networks and is used by many other works. The model uses an embedded function to encode each input to a lower dimensional vector and for each class c a prototype feature vector is defined as the mean vector of the embedded support data samples in this class, $v_c = \frac{1}{|S_c|} \sum_{(x_i y_i) \in S_c} f_\theta(x_i)$. An illustration of the procedure is presented in Figure 12. Prototypical Networks learn a metric space by computing distances of an embedded query point with each prototype representation with a softmax over the inverse of those distances. The authors claim that squared euclidean distance is a member of Bregman divergences functions for which have been proved that the point that minimizes the distance between all points is the mean of the cluster.

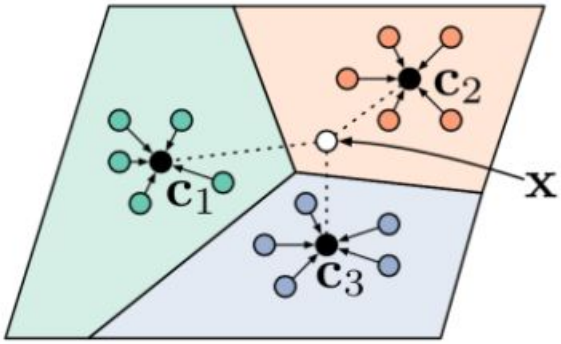

Figure 12: The prototypes for each class c_k are computed as the mean of embedded support examples for each class, (image from: Snell et al., 2017)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

A similar model is Relation Network proposed in (Sung et al., 2017).The model consist of two modules, the embedding module f_φ and the relation module g_φ. Samples from both query and support set, x_j and x_i respectively, are fed into the embedding module which produces the corresponding feature maps. The feature maps are combined through concatenation in depth and are fed into the relation module g_φ which produces a relation score representing the similarity between x_i and x_j. A visualization of the process is shown in Figure 13. The above network can be compared with Siamese networks with some differentiations in the way of capturing the relationships of the embedded vectors and the objective function the two methods are using. As mentioned before, Siamese networks compute L_1 distance between feature maps and the final loss is cross-entropy in order to get the final class prediction. In contrast, in Relation Networks, a convolutional classifier provides the relation between feature maps and minimum squared error (MSE) is used as the loss function.

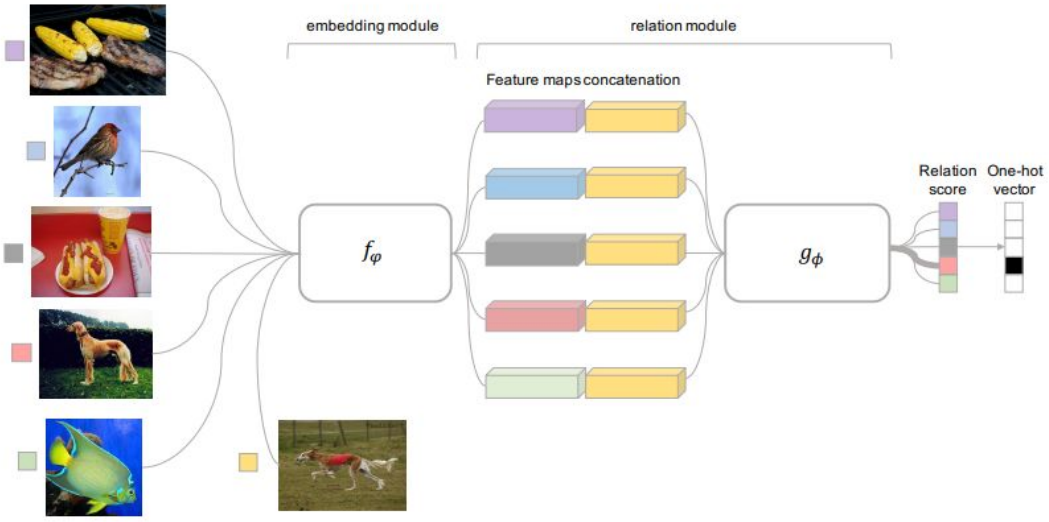

Figure 13: Relation Network architecture for a 5-way 1-shot problem with one query example, (image from: Sung et al., 2017)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

3 PUBLIC DATASETS FOR MODELS' TRAINING

REBUILD project's objective is to recognize the needs of refugees and migrants and provide them personalized assistance for a smooth integration at their destination states. The aim of WP3 is to provide a complete model that will help migrants and refugees to this goal. To this end, there must be a lot of profile data available to be used for the corresponding tasks, as profile modeling and profile analysis and integrally for the training of artificial intelligence models. On the one hand, those data should be representative and comprehensive regarding the services REBUILD application will provide and one the other hand they should comply to the GDPR's conditions for legal and ethical issues.

Regarding the constraints and the nature of needed data, it is obvious that it is very difficult to find available datasets that would help for the progress of WP3 tasks. Nevertheless, some open datasets can that be characterized as relative have been found during the research.

An extensive catalogue related to migration is the Information Catalogue of the Knowledge Centre on Migration and Demography[1] (KCMD) provided by the European Commission. This catalogue collects information from various web portals which are involved with migration issues such as Eurostat, ILO, OECD, UNICEF etc. The related issues have to do with legal migration and integration, children in migration, migrant smuggling, asylum and refugees and various demographics. Among them, one of the most important organizations is Eurostat[2] as it is the official statistical data provider of the European Union providing statistics for Europe. Although Eurostat has a wide variety of migration issues, the data it provides is in annual form for each European country so they can not be used for profile analysis and the corresponding tasks of WP3.

Another case that provides data per country and age group is the Suicide Rates Overview 1985 to 2016[3] constructed as a combination of other datasets provided by United Nations Development Program, World Bank and World Health Organization. The dataset contains 27800 records and 12 features including year, sex, age group, suicide rate etc.

The Skills For Jobs dataset provided by the Organisation for Economic Co-operation and Development [4] (OECD), connects the skills with the labour market. It provides country level information on the alignment between the demand and supply including cognitive, social and physical skills. The dataset includes 150 job-specific knowledge areas, skills and abilities for more than 40 countries. The dataset is used as a tool for discovering skills or changing careers. A dataset that is more close to the target, it is also provided by OECD and it is related with the Programme for the International Assessment of Adult Competencies[5] (PIAAC). It contains adults' measurements about education, social background, literacy, numeracy and ICT skills, language and information processing skills and how adults use those skills at home, at work and in the wider community. Thirty three countries participated in the program and 250.000 adults were surveyed in 87 different features. The advantage of this dataset is that it contains records per person but the deterrent to use it is that the statistical sample and the examined features have nothing to do with the scope of REBUILD project and the results would not be representative.

1 https://bluehub.jrc.ec.europa.eu/catalogues/data/
2 https://ec.europa.eu/eurostat/data/database
3 https://www.kaggle.com/russellyates88/suicide-rates-overview-1985-to-2016
4 https://www.oecdskillsforjobsdatabase.org/#EL/_
5 https://www.oecd.org/skills/piaac/

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

After an extensive research, the most relevant and in the most convenient form is ESCO dataset provided by European Commission [6]. ESCO data - European Skills, Competences, Qualifications and Occupations - is the European multilingual classification of the corresponding topics. ESCO provides a dictionary that defines, classifies and describes all the professional occupations within the European Union and their corresponding skills. It is also provides a comprehensive matching between occupations and their associated skills essential or optional. As it is pointed out the aim of ESCO is to support mobility across Europe for a more efficient labour market by offering a common terminology to the associated stakeholders.

ESCO provides 2942 occupations and 13485 skills in 27 languages. Except for all european languages plus Icelandic, Norwegian and Arabic are included. For each occupation there is information about the associated Isco Group, a preferred or alternative labels and a comprehensive text description. The same information there is for each skill plus the gradation of necessity. Both occupations and skills are divided into Isco Groups and Skill Groups respectively and they contain the high level categories for each topic. There are 10 Isco Groups (armed forces occupations, managers, professionals, technicians and associate professionals, clerical support workers, skilled agricultural forestry and fishery workers, craft and related trades workers, plant and machine operators and assemblers, elementary occupations) that contain all occupations and each occupation is assigned exclusively to one group. The skills are distinguished between skill/competence concepts and knowledge concepts. There is no distinction between skills and competences and there is not a full hierarchical structure among skill groups.

The dataset has been saved in Elasticsearch, a search and analytics engine which offers a dashboard builder named Kibana. Kibana is a visualization and management tool that provides real-time histograms, line graphs and pie charts. For a better display and a more easy exploration of the data three dashboards have been created. Each one presents information about occupations, skills and their relationships and through queries the user can find specific information. Some parts of the dashboards are shown in Figure 14, Figure 15, Figure 16.

6 https://ec.europa.eu/esco/portal/home

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

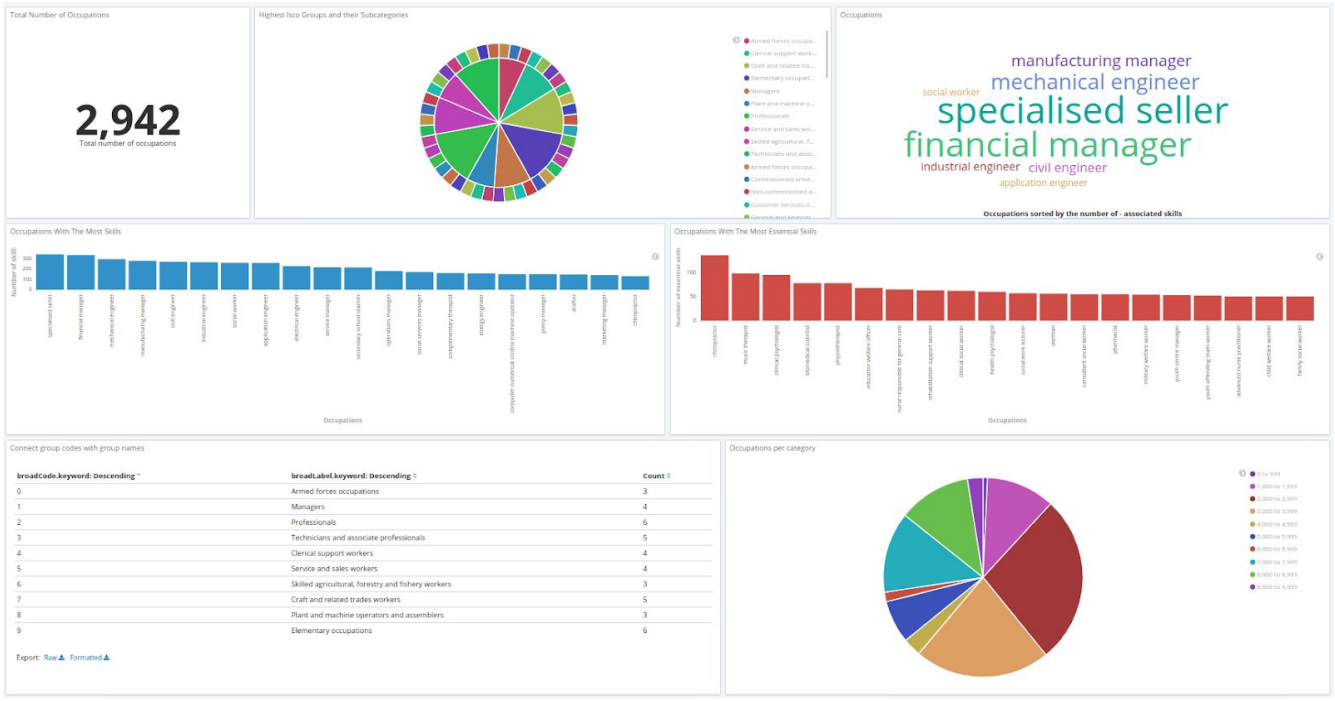

Figure 14: The total number of occupations, a pie chart with isco groups and their subcategories and a tag cloud with the occupations with the most skills are presented in the first row. In the second row there are two histograms with the occupations with the most and the most essential skills. For example the occupation that requires the most skills is "specialised seller" and it is associated with 340 skills, essential and optional. In the third row each isco group is associated with a code and the pie chart shows the total number of occupations per isco group. For example the biggest part in pie-chart is occupied from red region which represent the "Professionals" Isco group with 804 different occupations.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

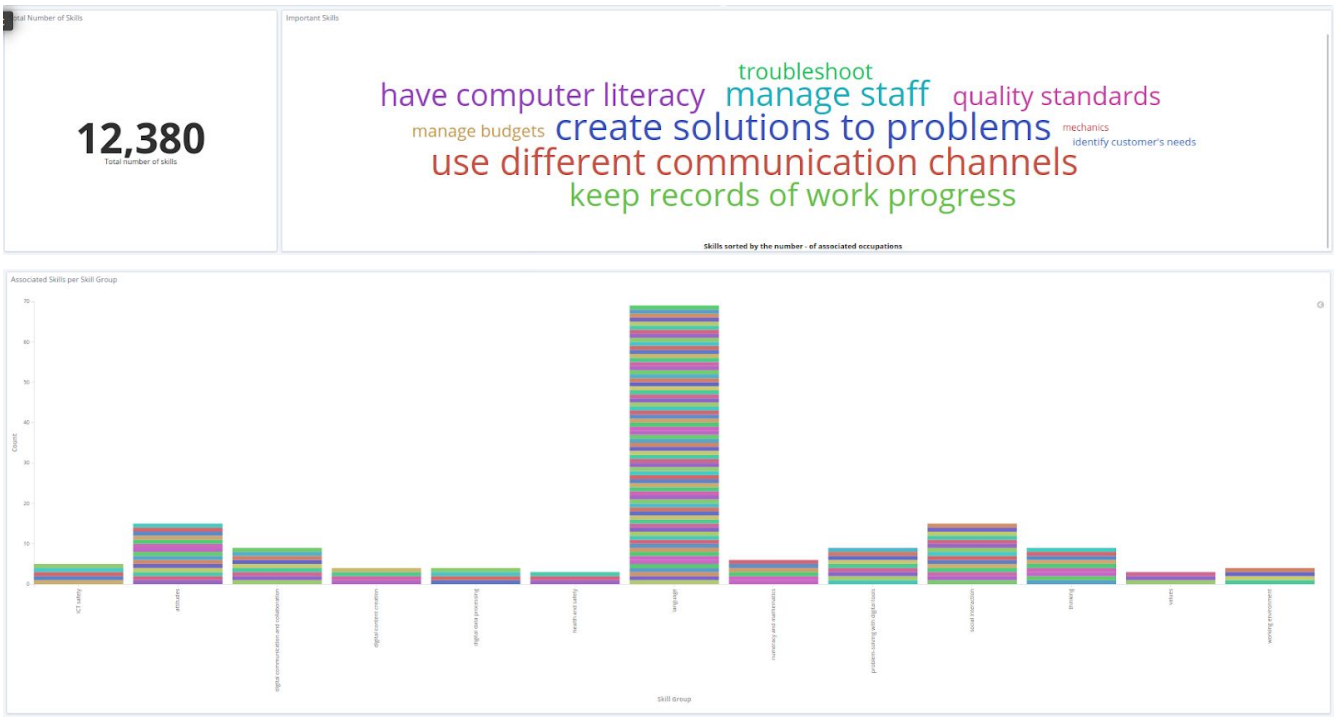

Figure 15: In the first row the total number of skills is mentioned and a tag cloud with the skills that are associated with the most occupations. In the second row there is a histogram that counts the sub categories per skill group. Each column is a different skill group and each different color in the columns represents a different skill. For example the skill group with the most subcategories corresponds to "Language" and the associated skills are different languages such as Greek, Spanish, Italian. It should be mentioned again that for the skill groups there is not a full hierarchical structure in contrast with the occupations' hierarchy.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

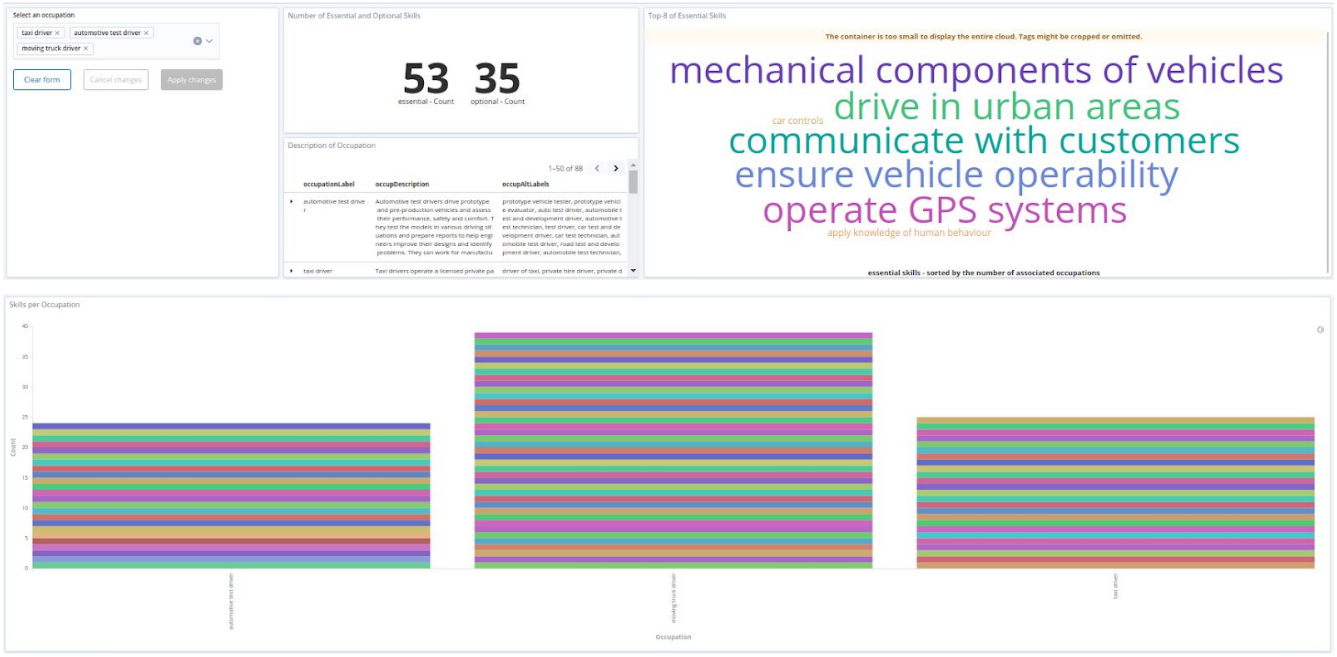

Figure 16: In this dashboard the user can select specific occupations and explore their relationships with the corresponding skills. In the specific example "taxi driver", "automotive test driver" and "moving truck driver" are selected. The total number of essential and optional skills is computed and the corresponding description with the alternative names for each occupation is presented. In the tag cloud there are the essential skills for the specific occupations sorted by the number of occupations they are associated. In the second row each column in the histogram is associated with each one of the selected occupations and includes the related skills. For example "moving truck driver" has the most skills which are 39 in total.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

4 METHODOLOGY

The proposed method for profile analysis is based on spectral clustering with deep learning techniques. The aim of the proposed method is to find a new embedded space where samples with the same characteristics have a similar or close representation. In this way, in the scope of clustering, similar samples can be collected in the same cluster while the dissimilar ones to different clusters. At the same time in the scope of retrieval, a new test sample can be transformed into the embedded representation which then can be associated with the most relative training samples. In the REBUILD application each sample represents a user profile with background information. The criterion of similarity, for example ethnicity, professional experience, religion etc. will be precisely defined in the next few months when the kind of data that will be used will also be defined through the co-creation workshops. Thus, taking into consideration the kind of data to be used and the exact purpose of the model, the procedure for the final output will be defined in the next deliverables of WP3.

In the proposed method, a deep neural network performs spectral clustering in the input dataset. It takes as input the training data points, transforms them into their spectral representation as low dimensional vectors and then k-means algorithm is applied to the embedded vectors for the final clustering. For the evaluation of the model, a test sample is transformed into the embedded space and the most relative training samples are provided as the final output, working as a recommendation framework. In the proposed method, Spectralnet (Shaham et al., 2018) is used as the baseline model but advanced techniques are used for the affinity matrix, in order to represent better the not obvious relations among input data and to combine more complex information.

As mentioned in a previous section Spectralnet is a deep learning approach for spectral clustering. It embeds the input data points into the eigenspace of their associated graph Laplacian matrix and it uses constraint stochastic optimization in order to be applied in large datasets. The constraints are implemented with a linear layer that force orthogonality in the network's output. After the completion of the training, a mapping function to the data points' spectral representation is provided and each point is associated with a cluster assignment. Furthermore, the embedded function can be applied in new unseen data points in order to get their spectral representation.

Regarding the learning of the mapping function, let w be a symmetric affinity function $R^d \times R^d \rightarrow [0,\infty)$ and $w(x,x')$ represent the similarity between data points x and x'. In order for the model to maintain the distances between the data points, the loss function can be defined as: $L(\theta) = E\left[w(x,x') \| y - y' \|^2\right]$, where y, y' is the output of SpectralNet for x, x' input respectively. An unwanted case to minimize the above function is by mapping all points to the same output vector y_0. In order to avoid this, the required constraint is the outputs to be orthogonal with respect to the inputs, so $E\left[yy^T\right] = I_{kxk}$ has to be satisfied.

The optimization procedure is performed in a stochastic way, by sampling a minibatch of m samples at each iteration and minimizing the loss function. The weights of the last linear layer used by the network are set by the QR decomposition of the input minibatch. The training of spectralNet is performed by alternation between orthogonalization and gradient steps. Each step uses a different minibatch, in orthogonalization step the weights of the last layer are set through QR decomposition and in the gradient step the remaining weights are set through backpropagation. Since the training of the model is completed the new embeddings of input data points emerge from a feedforward pass in SpectralNet. With this stochastic procedure, the model can be applied to large datasets and after the training an embedding function can be provided that maps new unseen data points

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

to its spectral embedding representation. The main difference with original spectral clustering is that the optimization is performed in stochastic way and makes an approximation of the true Laplacian eigenvectors. Then the k-means algorithm is applied in the new embedded data in order to get the cluster assignment for each data point.

Regarding the affinity matrix used by SpectralNet, it is constructed by a Siamese Network (Bromley et al., 1993) which is trained on unlabeled data and tries to learn the pairwise distances between data points. Siamese Networks are trained on positive and negative pairs of samples trying to learn affinity relations between them. In supervised learning this can be achieved by constructing pairs with the same or different label while in unsupervised learning the pairs labeling can be acquired by other methods such as graph distance or Euclidean proximity. The Siamese network used by SpectralNet is trained on positive-negative pairs that have been pre defined by the nearest neighbor algorithm. After the training of Siamese network, for each input data point x_i a new embedding vector $z_i = G_{\theta Siamese}(x_i)$ is provided. For each minibatch used by SpectralNet, Siamese network provides the embeddings z_i and the final affinity matrix W is constructed by replacing the Euclidean distance in equation (1) with $\| z_i - z_j \|$.

In the proposed methodology Siamese Networks are not used for the construction of the affinity matrix. Instead, an advanced affinity matrix is used to represent correlations among input data and to combine more complex information. The construction of affinity matrix is highly related with the input data, their form and the way they are pre-processed. In the absence of data that belongs to migrants' profiles and the fact that there is not a convenient dataset available, ESCO data are used by the proposed model. More specifically the text description of each occupation and skill and the matching among them are used in order to construct the advanced affinity matrix. The matching between occupations and skills is the kind of information tried to be captured by the proposed model, in order to be able to discover "hidden" relationships, and be capable to provide personalized recommendations to a user for possible jobs. The advanced affinity matrix can be considered as a "double affinity" with size nxn which contains the combination of two independent affinity matrices with the same size. The size n of the matrices is 2942, as is the number of occupations.

The first matrix is constructed with the information given by the number of common skills among occupations. More specifically, each occupation x is compared with all the other occupations $x_i,\ i = 1...n$ and for each pair, common skills between them is counted. Initially, common skills are computed as a scalar, so each pair (x_i, x_j) has equal skills as the pair (x_j, x_i). However, in the final affility the result has been normalized regarding the number of skills the examined occupation x has in total. Thus the affinity matrix is not symmetric since the proposed metric for pairs (x_i, x_j) and (x_j, x_i) is not the same. The computational cost of this procedure is $O(n^2)$, but it is computed just once as an external procedure, thus there is no additional cost during the use of the model.

The construction of the second affinity matrix is based on NLP (Natural Language Processing) preprocessing in occupations' text description. More specifically, a doc2vec model (Le, & Mikolov, 2014). is used, which is an extension of the very popular wordvec model proposed in (Mikolov, Chen, Corrado, & Dean, 2013). Those models take as input raw text corresponding to a word, paragraph or document and generate a vector representation. The algorithm those models use, exploits the information given by the context - the surrounding words of the examined word - and the goal is to represent semantically similar concepts with close vectors. An illustration of the described process is presented in Figure 17.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

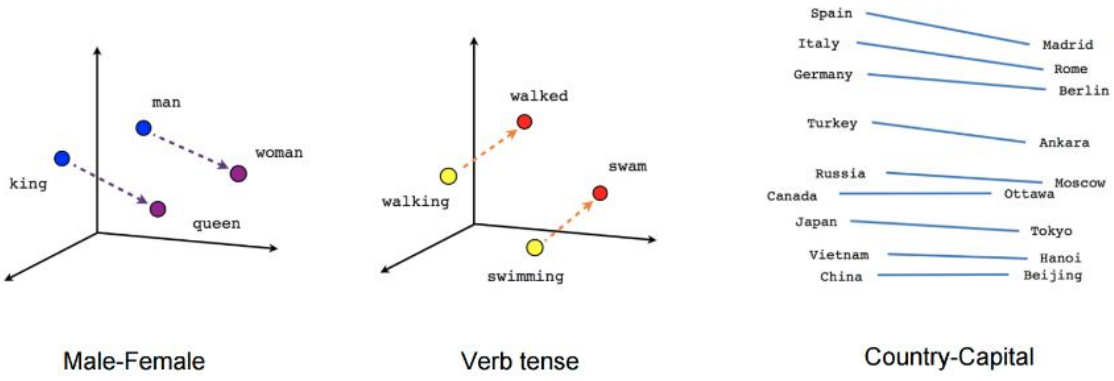

Figure 17: Visualization of word2vec functionality, image from (John, 2016)

In the proposed model, text from occupations' description is used to train a doc2vec model in order to get vector representations with 100 dimensions. With this procedure the goal is relative occupations to have close vector representations. Using vectors with more representative semantic information, clustering is assumed to be more efficient.
The second affinity matrix is constructed by measuring the cosine similarity among all occupation pairs as in the previous case. The affinity matrix is now symmetric and its size is nxn. As for the computational cost of measuring the pair distance is $O(n^2)$ but also it is computed as an external procedure without encumber the model's performance.

The final double affinity matrix used by SpectralNet, is the combination of the two independent matrices, by element wise addition to capture more information. SpectralNet approximates the true eigenvectors of the affinity matrix and after the completion of the training, it provides an embedding function and a cluster assignment for each input data point. The embedding function can also be used in unseen data points.

For the evaluation of the model the following procedure is performed:

1. The user gives a text input describing previous jobs he/she has made or relative professional experience.
2. The input text is used by doc2vec model in order to obtain the vector representation of the text.
3. This vector representation is given as input to SpectralNet's embedding function to get the final embedded vector.
4. The new embedded vector is compared with the already known embedded data points and the pairwise cosine similarity is computed. The corresponding occupations with the highest similarity is the final result and it works as a job recommendation, taking into account the user's previous professional experience.

In Figure 18 there is an illustration of the above process

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

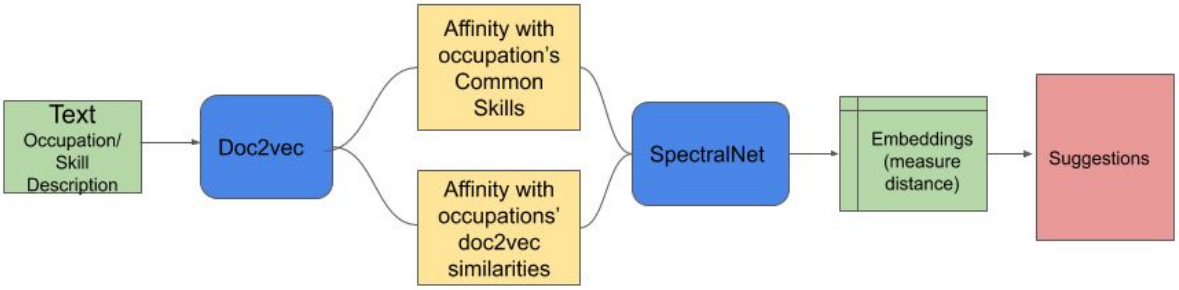

Figure 18: An illustration for the evaluation of the proposed method

In Table 1 some examples of the experimental results are presented. In the first column the input text describes some skills or a previous occupation of the user. As it seems, the proposed model actually learns "hidden" relationships among data. Exept for the expected results many occupations are connected in a not obvious way but the result is still make sense. In the "electrical installations" example the "electrical equipment assembler" or "Control panel tester" suggestions are assumed as very obvious answers while "quality engineer" can be considered as a more advanced suggestion. Apparently among suggestion there are cases that can be described as irrelevant such as "palliative care social worker" in "I know programming and how to use computers" example. It is important to note those results emerge from a first approach in training on the available dataset. Further training and tuning has to be done both on the available dataset but mainly in a project specific dataset that contains user profiles in order to achieve the REBUILD project's objective.

Input text	Suggested occupations	
i know how to make electrical installations	telecommunications equipment specialised seller	leather goods machine operator
	import export manager in electrical household appliances	computer and accessories specialised seller
	wholesale merchant in mining, construction and civil engineering machinery	fluid power technician
	technical sales representative in electronic and telecommunications equipment	control panel tester
	electrical equipment assembler	quality engineer
I had animals cows, horses, sheep	housekeeping supervisor	import export specialist in dairy products and edible oils
	import export manager in flowers and plants	cocoa press operator
	groom	underground heavy equipment operator

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

	grill cook	wholesale merchant in waste and scrap
	pig breeder	wholesale merchant in flowers and plants
I know programming and how to use computers	ICT security administrator	economic development coordinator
	ICT network architect	ICT resilience manager
	user interface designer	journalism lecturer
	ICT network engineer	palliative care social worker
	columnist	gambling, betting, and lottery game developer
I taught children in school	literature teacher secondary school	screw machine operator
	home care aide	laser cutting machine operator
	classical languages teacher secondary school	philosophy teacher secondary school
	religious education teacher secondary school	snowboard instructor
	mathematics teacher secondary school	drilling machine operator
I was a farmer and had agricultural works	viticulture adviser	import export specialist in china and other glassware
	textile printer	material stress analyst
	import export specialist in chemical products	arboriculturist
	crop production manager	wholesale merchant
	agriculture, forestry and fishery vocational teacher	dietitian

Table 1 – Experimental results

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

5 CONCLUSION

This deliverable is a report about the progress of REBUILD project's WP3 and specifically about the task 3.1. This task includes the user's profile analysis and modeling from raw data to representative vectors. These vectors will be used as input by the AI-based models in order to provide personalized suggestions and recommendations to the final users. An important part of 3.1 task is the use of embedded methods in order to create low dimensional vectors that will be used for the different personalization and matching features.

In the first part of deliverable a comprehensive presentation of the current state of the art methods is presented. These methods can handle data in many formats, transform them into a more representative way and provide models that can be used for clustering, classification or retrieval depending on the exact purpose of the final model. Taking into consideration the absence of data that belongs to migrants' profiles and the fact that there is not a similar dataset available, in the next part of the deliverable a discussion about the available datasets that could be used for the training of AI models is performed. Using an available dataset and some interesting methods, a first approach for the creation of the necessary model for WP3 has been made. In the final part of the deliverable a detailed presentation of the proposed methodology is performed.

It is important to note that within the co-creation workshops the kind and the form of input data as the actual needs of refugees and the final personalized services REBUILD application will provide is expected to be clearly defined. To this end, this deliverable does not aim to analyze what data will be used, as to present an analysis on how the profile modeling can be performed.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

6 REFERENCES

Ali, T., Asghar, S., & Naseer Ahmed Sajid. (2010). Critical analysis of DBSCAN variations. *2010 International Conference on Information and Emerging Technologies*, 1–6. https://doi.org/10.1109/ICIET.2010.5625720

Andrew, G., Arora, R., Bilmes, J., & Livescu, K. (2013). Deep Canonical Correlation Analysis. *International Conference on Machine Learning*, 1247–1255. Retrieved from http://proceedings.mlr.press/v28/andrew13.html

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2006, December 4). *Greedy layer-wise training of deep networks*. 153–160. Retrieved from http://dl.acm.org/citation.cfm?id=2976456.2976476

Bromley, J., Bentz, J., Bottou, L., Guyon, I., Lecun, Y., Moore, C., … Shah, R. (1993). Signature Verification using a "Siamese" Time Delay Neural Network. *International Journal of Pattern Recognition and Artificial Intelligence*, *7*, 25. https://doi.org/10.1142/S0218001493000339

Cao, F., Estert, M., Qian, W., & Zhou, A. (2006). Density-Based Clustering over an Evolving Data Stream with Noise. *Proceedings of the 2006 SIAM International Conference on Data Mining*, 328–339. https://doi.org/10.1137/1.9781611972764.29

Chen, Z.-M., Wei, X.-S., Wang, P., & Guo, Y. (2019). Multi-Label Image Recognition with Graph Convolutional Networks. *ArXiv:1904.03582 [Cs]*. Retrieved from http://arxiv.org/abs/1904.03582

Chidananda Gowda, K., & Krishna, G. (1978). Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, *10*(2), 105–112. https://doi.org/10.1016/0031-3203(78)90018-3

Corpet, F. (1988). Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Research*, *16*(22), 10881–10890. https://doi.org/10.1093/nar/16.22.10881

Ding, C., & Xiaofeng He. (2002). Cluster merging and splitting in hierarchical clustering algorithms. *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, 139–146. https://doi.org/10.1109/ICDM.2002.1183896

Dizaji, K. G., Herandi, A., Deng, C., Cai, W., & Huang, H. (2017). Deep Clustering via Joint Convolutional Autoencoder Embedding and Relative Entropy Minimization. *ArXiv:1704.06327 [Cs]*. Retrieved from http://arxiv.org/abs/1704.06327

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd (Vol. 96, No. 34, pp. 226-231).

Finn, C., Abbeel, P., & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *ArXiv:1703.03400 [Cs]*. Retrieved from http://arxiv.org/abs/1703.03400

Gheche, M. E., Chierchia, G., & Frossard, P. (2018). OrthoNet: Multilayer Network Data Clustering. *ArXiv:1811.00821 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1811.00821

Gidaris, S., & Komodakis, N. (2018). Dynamic Few-Shot Visual Learning without Forgetting. *ArXiv:1804.09458 [Cs]*. Retrieved from http://arxiv.org/abs/1804.09458

Gong, Y., Jia, Y., Leung, T., Toshev, A., & Ioffe, S. (2013). Deep Convolutional Ranking for Multilabel Image Annotation. *ArXiv:1312.4894 [Cs]*. Retrieved from http://arxiv.org/abs/1312.4894

Graves, A., Wayne, G., & Danihelka, I. (2014). Neural Turing Machines. *ArXiv:1410.5401 [Cs]*. Retrieved from http://arxiv.org/abs/1410.5401

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. 770–778. Retrieved from http://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html

Hinton, G. E., & Plaut, D. C. (1987). Using Fast Weights to Deblur Old Memories. *In Proceedings of the 9th Annual Conference of the Cognitive Science Society*, 177–186. Erlbaum.

Hochreiter, S., Younger, A. S., & Conwell, P. R. (2001). Learning to Learn Using Gradient Descent. In G. Dorffner, H. Bischof, & K. Hornik (Eds.), *Artificial Neural Networks—ICANN 2001* (pp. 87–94). Springer Berlin Heidelberg.

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). *Densely Connected Convolutional Networks*.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

4700–4708. Retrieved from http://openaccess.thecvf.com/content_cvpr_2017/html/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.html

Huang, S., Ota, K., Dong, M., & Li, F. (2019). MultiSpectralNet: Spectral Clustering Using Deep Neural Network for Multi-View Data. *IEEE Transactions on Computational Social Systems*, *6*(4), 749–760. https://doi.org/10.1109/TCSS.2019.2926450

Jeong, Y., Lee, S., Park, D., & Park, K. H. (2018). Accurate Age Estimation Using Multi-Task Siamese Network-Based Deep Metric Learning for Frontal Face Images. *Symmetry*, *10*(9), 385. https://doi.org/10.3390/sym10090385

Jianbo Shi, & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905. https://doi.org/10.1109/34.868688

John, V. (2016). Rapid-Rate: A Framework for Semi-supervised Real-time Sentiment Trend Detection in Unstructured Big Data. *ArXiv:1703.08088 [Cs]*. https://doi.org/10.13140/RG.2.2.32385.04966

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, *32*(3), 241–254. https://doi.org/10.1007/BF02289588

Karypis, G., Eui-Hong Han, & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, *32*(8), 68–75. https://doi.org/10.1109/2.781637

Koch, G. R. (2015). *Siamese Neural Networks for One-Shot Image Recognition*.

Kriegel, H.-P., Kröger, P., Sander, J., & Zimek, A. (2011). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(3), 231–240. https://doi.org/10.1002/widm.30

Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188-1196).

Liu, J., Chang, W.-C., Wu, Y., & Yang, Y. (2017). Deep Learning for Extreme Multi-label Text Classification. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 115–124. https://doi.org/10.1145/3077136.3080834

MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations*. Presented at the Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. Retrieved from https://projecteuclid.org/euclid.bsmsp/1200512992

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Munkhdalai, T., & Yu, H. (2017). Meta Networks. *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 2554–2563. Retrieved from http://dl.acm.org/citation.cfm?id=3305890.3305945

Nam, J., Kim, J., Loza Mencía, E., Gurevych, I., & Fürnkranz, J. (2014). Large-Scale Multi-label Text Classification—Revisiting Neural Networks. In T. Calders, F. Esposito, E. Hüllermeier, & R. Meo (Eds.), *Machine Learning and Knowledge Discovery in Databases* (pp. 437–452). Springer Berlin Heidelberg.

Ng, A. (2011). Sparse autoencoder. CS294A Lecture notes, 72(2011), 1-19.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On Spectral Clustering: Analysis and an Algorithm. *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, 849–856. Retrieved from http://dl.acm.org/citation.cfm?id=2980539.2980649

Nichol, A., Achiam, J., & Schulman, J. (2018). On First-Order Meta-Learning Algorithms. *ArXiv:1803.02999 [Cs]*. Retrieved from http://arxiv.org/abs/1803.02999

Qi, H., Brown, M., & Lowe, D. G. (2017). Low-Shot Learning with Imprinted Weights. *ArXiv:1712.07136 [Cs]*. Retrieved from http://arxiv.org/abs/1712.07136

Ravi, S., & Larochelle, H. (2016). *Optimization as a Model for Few-Shot Learning*. Retrieved from https://openreview.net/forum?id=rJY0-Kcll

Sander, J., Ester, M., Kriegel, H., & Xu, X. (2010.). *Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications*.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., & Lillicrap, T. (2016). Meta-learning with Memory-augmented Neural Networks. *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, 1842–1850. Retrieved from http://dl.acm.org/citation.cfm?id=3045390.3045585

Shah, S. A., & Koltun, V. (2017). Robust continuous clustering. *Proceedings of the National Academy of*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 822215

Sciences, *114*(37), 9814–9819. https://doi.org/10.1073/pnas.1700770114
Shah, S. A., & Koltun, V. (2018). Deep Continuous Clustering. *ArXiv:1803.01449 [Cs]*. Retrieved from http://arxiv.org/abs/1803.01449
Shaham, U., Stanton, K., Li, H., Nadler, B., Basri, R., & Kluger, Y. (2018). SpectralNet: Spectral Clustering using Deep Neural Networks. *ArXiv:1801.01587 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1801.01587
Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
Snell, J., Swersky, K., & Zemel, R. S. (2017). Prototypical Networks for Few-shot Learning. *ArXiv:1703.05175 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1703.05175
Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S., & Hospedales, T. M. (2017). Learning to Compare: Relation Network for Few-Shot Learning. *ArXiv:1711.06025 [Cs]*. Retrieved from http://arxiv.org/abs/1711.06025
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). *Rethinking the Inception Architecture for Computer Vision*. 2818–2826. Retrieved from https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.html
Tzoreff, E., Kogan, O., & Choukroun, Y. (2018). Deep Discriminative Latent Space for Clustering. *ArXiv:1805.10795 [Cs]*. Retrieved from http://arxiv.org/abs/1805.10795
Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *J. Mach. Learn. Res.*, *11*, 3371–3408.
Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching Networks for One Shot Learning. *ArXiv:1606.04080 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1606.04080
von Luxburg, U. (2007). A Tutorial on Spectral Clustering. *ArXiv:0711.0189 [Cs]*. Retrieved from http://arxiv.org/abs/0711.0189
Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *37*(3), 328–339. https://doi.org/10.1109/29.21701
Wang, J., Hilton, A., & Jiang, J. (2019). Spectral Analysis Network for Deep Representation Learning and Image Clustering. *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 1540–1545. https://doi.org/10.1109/ICME.2019.00266
Wang, Jiang, Yang, Y., Mao, J., Huang, Z., Huang, C., & Xu, W. (2016). CNN-RNN: A Unified Framework for Multi-label Image Classification. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2285–2294. https://doi.org/10.1109/CVPR.2016.251
Wang, Z., Chen, T., Li, G., Xu, R., & Lin, L. (2017). *Multi-Label Image Recognition by Recurrently Discovering Attentional Regions*. 464–472. Retrieved from http://openaccess.thecvf.com/content_iccv_2017/html/Wang_Multi-Label_Image_Recognition_ICCV_2017_paper.html
Weston, J., Chopra, S., & Bordes, A. (2014). Memory Networks. *ArXiv:1410.3916 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1410.3916
Wichrowska, O., Maheswaranathan, N., Hoffman, M. W., Colmenarejo, S. G., Denil, M., de Freitas, N., & Sohl-Dickstein, J. (2017). Learned Optimizers that Scale and Generalize. *ArXiv:1703.04813 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1703.04813
Xie, J., Girshick, R., & Farhadi, A. (2015). Unsupervised Deep Embedding for Clustering Analysis. *ArXiv:1511.06335 [Cs]*. Retrieved from http://arxiv.org/abs/1511.06335
Yang, B., Fu, X., Sidiropoulos, N. D., & Hong, M. (2016). Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering. *ArXiv:1610.04794 [Cs]*. Retrieved from http://arxiv.org/abs/1610.04794
Yang, J., Parikh, D., & Batra, D. (2016). Joint Unsupervised Learning of Deep Representations and Image Clusters. *ArXiv:1604.03628 [Cs]*. Retrieved from http://arxiv.org/abs/1604.03628
Yang, X., Deng, C., Zheng, F., Yan, J., & Liu, W. (2019). *Deep Spectral Clustering Using Dual Autoencoder Network*. 10.
Zhao, J., Xie, X., Xu, X., & Sun, S. (2017). Multi-view learning overview: Recent progress and new challenges.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

Information Fusion, *38*, 43–54. https://doi.org/10.1016/j.inffus.2017.02.007

Zhu, F., Li, H., Ouyang, W., Yu, N., & Wang, X. (2017). *Learning Spatial Regularization With Image-Level Supervisions for Multi-Label Image Classification*. 5513–5522. Retrieved from http://openaccess.thecvf.com/content_cvpr_2017/html/Zhu_Learning_Spatial_Regularization_CVPR_2017_paper.html

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215

REBUILD

ICT-enabled integration facilitator and life rebuilding guidance

Deliverable: D3.1 Users' profile modeling

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 822215.