MIC: Multi-view Image Classifier using Generative Adversarial Networks for Missing Data Imputation

Conference Paper · March 2021

CITATIONS
0

READS
2

5 authors, including:

Mahmoud Jarraya
EURA NOVA
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Maher Marwani
EURA NOVA
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Gianmarco Aversano
EURA NOVA
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Ichraf Lahouli
Euranova
12 PUBLICATIONS **12** CITATIONS
SEE PROFILE

MIC: Multi-view Image Classifier using Generative Adversarial Networks for Missing Data Imputation

1st Gianmarco Aversano
EURANOVA BE
Mont-Saint-Guibert, Belgium
gianmarco.aversano@euranova.eu

2nd Mahmoud Jarraya
EURANOVA TN
Les berges du lac, Tunisia
mahmoud.jarraya@euranova.eu

3rd Maher Marwani
EURANOVA TN
Les berges du lac, Tunisia
maher.marwani@euranova.eu

4th Ichraf Lahouli
EURANOVA TN
Les berges du lac, Tunisia
ichraf.lahouli@euranova.eu

5th Sabri Skhiri
EURANOVA BE
Mont-Saint-Guibert, Belgium
sabri.skhiri@euranova.eu

***Abstract*—In this paper, we propose a framework for image classification tasks, named MIC, that takes as input multi-view images, such as RGB-T images for surveillance purposes. We combine auto-encoder and generative adversarial network architectures to ensure the multi-view embedding in a common latent space. Then, the resulting features are fed to the classification stage. The proposed framework is able to, all at once, train the multi-view embedding model to find a shared latent representation for the different views, perform data imputation (generate the missing views) and ensure the classification task by predicting the labels. Experiments on the MNIST dataset with a panoply of classifiers and several missingness ratios show the effectiveness of our solution.**

I. INTRODUCTION

In real world scenarios, multi-view or multi-modal data are quite common as a sample can have different representations depending on its source, on the sensor that captured it, or even on the applied method generating its features. Consequently, multi-view images, which are often complementary, are exploited in various computer vision applications like image translation [1], breast cancer detection [2] or also surveillance applications where RGB and depth/thermal images are used for human action recognition and pedestrian detection [3], [4]. In agriculture, multi-spectral images combining the traditional RGB bands and the near-infrared band allow the prediction of some mineral concentrations like in [5]. The use of multi-view images even reaches the fascinating astrophysics domain where the mass of a galaxy cluster can be inferred based on X-ray and SZ images [6].

The shared challenge between all the aforementioned applications is to better exploit the complementary information and the similarities between the different views that refer to the same item. One popular approach, called Multi-View Embedding (MVE), is to project these views in a common latent space like shown in Figure 1. The learnt common latent manifold can then be used to solve several machine learning tasks such as clustering and classification.

The second challenge is related to missing data imputation. In real world, one or more views can be missing due to malfunction of a sensor or latency during the streaming phase. Several techniques have been used to handle multi-view datasets. Recently, Generative Adversarial Networks (GAN)s [9] have gained popularity in the field of MVE because of their ability to map any source distribution to a target one (e.g. the distribution of the missing data), which is often used to impute the missing data. For example, in [10], [11], high-level features extracted from each view are embedded and fused to provide a unique representation of the given sample and generate missing data by leveraging cycle consistency between the generators.

The main contribution of this paper is to propose a multi-view image classifier that is able to tackle the challenge of missing views even at a high ratio (with respect to the total amount of data). Indeed, the proposed framework is able to extract visual features from the multi-view images and embed these features in a common latent space. Thus, the classifier can predict the label based on this complementary information.

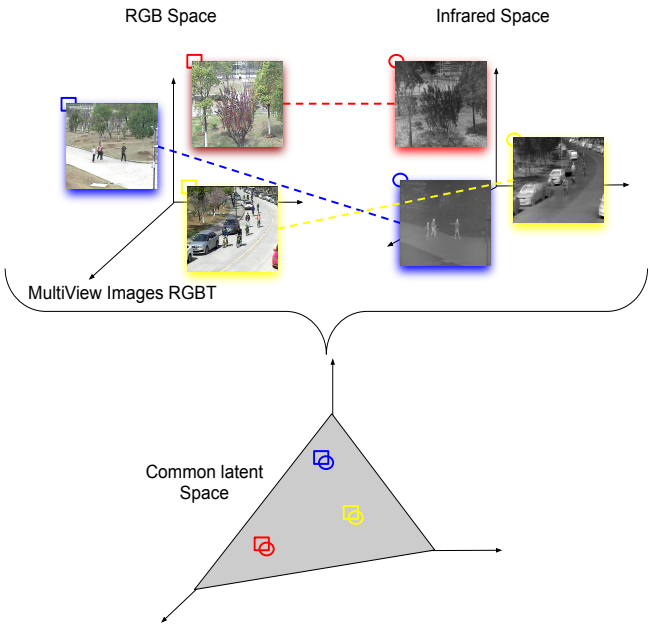


Fig. 1. Schematic of the Multi-View Embedding (MVE) to a common latent space (adapted from [7]). The images are taken from the VOT-RGBT 2020 challenge [8].

The remaining sections are organised as follows. First, an overview of the state-of-the-art research works concerning MVE techniques is exposed. Then, the proposed methodology is explained in details. Finally, experiments on the MNIST dataset with two views are conducted and comparison results with the state-of-the-art method are shown in order to validate the concrete contribution of this work.

II. RELATED WORKS

Multi-view representation learning is concerned with the problem of learning common (latent) representations of multi-view data. It is usually done by either aligning or fusing view-specific latent features. The obtained features are exploited for several applications such as cross-media retrieval, translation, video captioning and recommendation systems. However, the remaining challenge is how to deal with missing views. Mapping different views to a common latent space, instead of concatenating, facilitates the process of data imputation. Typical techniques include kernel-based methods such as Deep Canonical Correlation Analysis (DCCA) [12] or non-negative matrix factorization (NMF)-based methods [13]. However, these two kinds of techniques suffer from several limitations, mainly complexity, inefficiency to scale with large scale databases and inability to compensate for missing views due to regularization and added constraints [10] which makes them less attractive. Recently, generative models such as VAEs [14] or GANs have gained popularity in this field [10], [11]. A recent survey [7] reports the advantages of GAN-based techniques by narrowing the difference between the distributions of different modalities/views. For instance, the implementation of a multi-modal adversarial representation network (MARN) with an attention mechanism has led to state-of-the-art performance for click-through rate prediction [15]. For these reasons, the present work proposes a GAN-based architecture for MVE, inspired by [10], [11], that is able to impute missing views from a fused latent common space. Actually, we minimize the reconstruction loss to tune the fusion layer, differently from GPMVC [10] which uses a dedicated loss for it. Besides, unlike [10], our model does not make use of a clustering layer. We show that the performance is not affected and the classification accuracy even increases in most cases.

III. PROPOSED METHODOLOGY

The ultimate goal of this work is to solve an end-to-end multi-view classification problem. Thus, our solution consists of two main components. The first component is to ensure the MVE while the second one is to solve the classification task. This section mainly focuses on the first component by explaining the MVE model architecture, illustrated in Figure 2.

A. Notation

In this section, we first formalise the input data to be used for the rest of the paper. The multi-view data is represented by: $\mathbf{X} = \left\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(V)}\right\}$, where $\mathbf{X}^{(v)} = \left\{x_1^{(v)}, x_2^{(v)}, \cdots, x_N^{(v)}\right\} \in \mathbf{R}^{N \times d_v} \quad \forall v \in [1, V]$, V is the number of views, N is the number of samples, and d_v is the feature dimension of the v-th view. Besides, since our data might have missing views that need to be imputed, we split our data into 2 subsets: paired data $\left\{x^{(1)}, x^{(2)}, \cdots, x^{(V)}\right\}$, in which all views exist, and unpaired data, $\left\{x'^{(1)}, x^{(2)}, \cdots, x'^{(V)}\right\}$, in which at least one view is missing. We denote by $r \in [0, 1]$ the missing data ratio which means the percentage of the missing data with respect to the total amount of data. For instance, if $r = 0.1$, 10% of the samples have at least one missing view. We denote by $\mathcal{Y}$ the target space and each $x_i^{(v)}$ $\forall i \in [1, N]$ and $\forall v \in [1, V]$ is associated to $y_i \in \mathcal{Y}$.

B. Architecture

The purpose of the MVE component, denoted by $\mathcal{M}$, is to derive a representative feature vector in a common latent space of the multi-view input data. This can be achieved by a mixture of Auto-Encoder (AE) and GAN architectures. The MVE component architecture, illustrated in Figure 2, consists of four types of networks: Encoder E, Fusion layer F, Decoder G and a Discriminator D.

Encoder $\{E_v\}_{v=1}^{V}$: $R^{d_v} \rightarrow R^d$.

V encoders, each one projecting one input view $X^{(v)}$ to the corresponding view-specific subspace through many stacked convolution-batch normalization-ReLu layers. The output of each encoder is a d-dimensional vector $Z^{(v)}$.

Fusion F: $R^{d \times V} \rightarrow R^d$.

The objective of the fusion network is to capture the shared semantics of the multi-view data from the resulting V view-specific representation vectors. It takes the output of the encoders as input and derives the target feature vector through a fully-connected layer.

Decoder $\{G_v\}_{v=1}^{V}$: $R^d \rightarrow R^{d_v}$.

Each decoder G_v has a mirrored architecture to the corresponding E_v. It takes a vector from latent space as input and generates the corresponding view. It functions both as a decoder and a generator.

Discriminator $\{D_v\}_{v=1}^{V}$: $R^{d_v} \rightarrow \{0, 1\}$.

Each discriminator D_v takes a sample from its corresponding view distribution as input, which can either be the sample $\tilde{x}_i^{(v)}$ generated by G_v or the true sample $x_i^{(v)}$, and outputs the probability for this sample to be real, using a stack of convolution-batch normalization-LeakyReLu layers and a sigmoid activation.

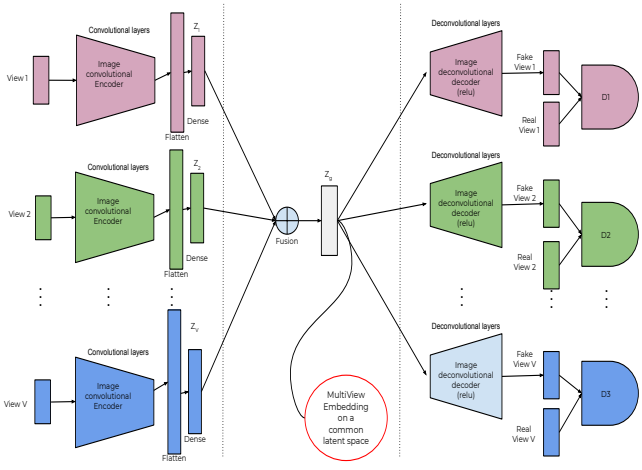


Fig. 2. GAN-based MultiView Embedding Architecture

C. Objective Function

The objective function that we optimize includes two terms: the auto-encoder loss and the GAN loss.

1) Auto-Encoder loss: This loss guarantees that the multi-view feature vector is representative enough of the original data and holds cross-view information. Thus, this loss is used to train the encoder, decoder and fusion networks, differently from [10] where a different loss is defined to train the fusion network. In practice, the fusion is performed only after a predefined number of epochs (fusion mode) of the training that minimizes the reconstruction loss defined as follows:

$$L_{rec} = \Sigma_{v=1}^{V} \|X^{(v)} - G_v(Z_v)\|^2 \tag{1}$$

with:

$$Z_v = \begin{cases} E_v(X^{(v)}), & \text{if } \textbf{not} \text{ in Fusion Mode} \\ F(E_1(X^{(1)}), \ldots, E_v(X^{(V)})) & \text{otherwise.} \end{cases} \tag{2}$$

Additionally, each decoder $\{G_v\}_{v=1}^{V}$ is expected to generate its corresponding view from a latent vector, regardless of whether it comes from a view-specific latent space or from the common (fused) latent space. Thus, we promote the view-specific latent vectors to be similar, by adding the alignment loss as follows:

$$L_{align} = \Sigma_{v=1}^{V} \Sigma_{v\prime>v}^{V} \|E_v(X^{(v)}) - E_{v\prime}(X^{(v\prime)})\|^2 \tag{3}$$

Therefore, the final Auto-Encoder loss is:

$$L_{AE} = L_{rec} + a * L_{align} \tag{4}$$

with $a \geq 0$ being a hyperparameter.

The views are reconstructed by the decoders, which take the common latent vector resulting from the fusion operation as input. Therefore, the training of the Auto-Encoder on minimizing L_{AE} is not possible in the case of unpaired data where fusion cannot be realized. To tackle the problem of missing data, we make use of adversarial training.

2) Adversarial Training loss: We employ the adversarial training (AT) loss to train the GAN part of the model, which is composed of the decoders $\{G_v\}_{v=1}^{V}$ and the discriminators $\{D_v\}_{v=1}^{V}$.

The goal of each $\{G_v\}_{v=1}^{V}$ is to fool the corresponding $\{D_v\}_{v=1}^{V}$ by producing realistic images from the latent vectors of the common latent space distribution, whereas

the goal of $\{D_v\}_{v=1}^V$ is to distinguish whether the input is real or fake. The GAN loss is expressed as follows:

$$L_{GAN} = \Sigma_{v=1}^{V}[\log D_v(X^{(v)}) + \log(1 - D_v \circ G_v(Z_v))] \quad (5)$$

In fact, the generators learn to map a given latent vector to a sample from its target distribution. Yet, the generated sample should be paired with the existing one. This cannot be ensured by relying only on the GAN loss. Therefore, we add the cycle loss defined in Eq. (6). This way, $\{G_v\}_{v=1}^V$ are trained also on the imputed data from the existing one.

$$L_{cyc} = \Sigma_{v=1}^{V}\Sigma_{v\prime\neq v}^{V}\|X^{(v)} - G_v \circ E_{v\prime} \circ G_{v\prime} \circ E_v(X^{(v)})\| \quad (6)$$

The final adversarial training loss is:

$$L_{AT} = L_{cyc} + L_{GAN} \quad (7)$$

D. Implementation

The training consists of a two-stage procedure where we first pre-train the AE part. Then, we pre-train the GAN part.

Step 1: The paired data is loaded to $\{E_v\}_{v=1}^V$ to get V vectors in the latent space. First, these vectors are passed directly to their corresponding $\{G_v\}_{v=1}^V$ to reconstruct the input. This way, by minimizing Eq. (3), the different vectors are forced to be close to each other in the view-specific latent spaces. Then, after a portion ϕ of the total number of epochs, the V vectors are fused by F before being passed to $\{G_v\}_{v=1}^V$ (Fusion Mode). Finally, we update the networks by minimizing Eq. (4).

Step 2: At this stage, we use the pre-trained $\{E_v\}_{v=1}^V$, $\{G_v\}_{v=1}^V$ and F in step 1 to impute the missing data. After that, we train the GAN part by minimizing L_{AT} loss defined in Eq. (7).

The training procedure is detailed in Algorithm 1.

Input: X, ϕ, number of epochs N_{AE}, N_{GAN}
Output: MVE model $\mathcal{M}$
Initialize: E_v, F, G_v and D_v, $v \in [1, V]$.
step 1: Pre-train AE networks on paired data
for $iter \leq N_{AE}$ **do**
  Load **only paired** data $\{x^{(1)}, x^{(2)}, \cdots, x^{(V)}\}$;
  Compute $\mathbf{Z_v}$ by Eq. (2);
  if $iter < \phi * N_{AE}$ *with* $0 < \phi < 1$ **then**
    Update $\{E_v\}_{v=1}^V$ and $\{G_v\}_{v=1}^V$ by Eq. (4);
  else
    Update $\{E_v\}_{v=1}^V$, $\{G_v\}_{v=1}^V$ and F by Eq. (4);
  end
end
step 2: Pre-train GAN networks on all data
for $iter \leq N_{GAN}$ **do**
  Load **all** data X;
  Impute missing views;
  if $iter < \phi * N_{GAN}$ *with* $0 < \phi < 1$ **then**
    Update $\{G_v\}_{v=1}^V$ and $\{D_v\}_{v=1}^V$ by Eq. (7);
  else
    Update $\{G_v\}_{v=1}^V$, $\{D_v\}_{v=1}^V$ and F by Eq. (7);
  end
end

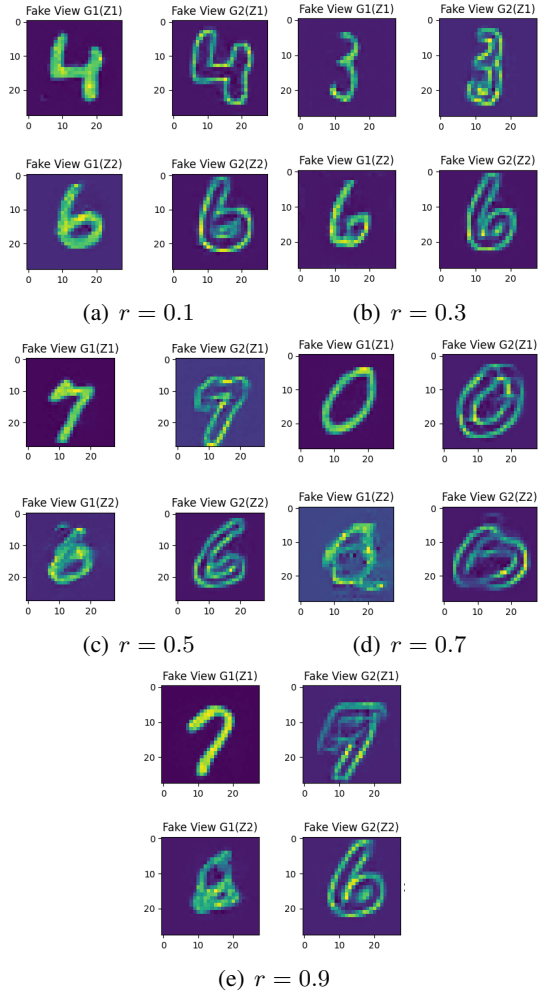
Algorithm 1: MVE model $\mathcal{M}$ training procedure

IV. EXPERIMENTS AND RESULTS

In order to validate the methodology proposed in this work, we evaluate the GAN-based MVE architecture for a classic classification task using the MNIST dataset. We consider the original digit images as the first view while we compute the corresponding edge images to simulate the second view like in [10], [11].

First, we validate the proposed MVE architecture and precisely the data imputation functionality in a qualitative manner by visualizing some of the generated samples. To do this, we run our model $\mathcal{M}$ five times with the same configuration except for the missing data ratio r in training and testing sets. At each run, we increase r by 0.2 starting from 0.1 and observe the generated views i.e the ones that were missing. We reveal some samples of the generators' output in the five experiments in Figure 3. We notice that the generations are quite realistic for the two first runs ($r = 0.1$ and $r = 0.3$). Yet, when more than half of the data is missing, the quality of the generations starts decreasing, which should affect the classification accuracy later on. For instance, the generated images get blurred when the missing data ratio r is equal to 0.9.

Secondly, we quantitatively test the robustness of the MVE component to deal with missing data by measuring the classification accuracy. First, we use the trained $\mathcal{M}$ in the previous experiments and extract the feature representations in the common latent space for both training and testing sets. Then, we use these feature vectors as input for a bunch of common classification models. Therefore, we evaluate the proposed model $\mathcal{M}$, with respect to the missingness ratio r, on an image classification task. The classification accuracy results are reported in Figure 4. Despite the fact that the classifiers have not been tuned, most of them achieved over 92%



(a) $r = 0.1$ (b) $r = 0.3$

(c) $r = 0.5$ (d) $r = 0.7$

(e) $r = 0.9$

Fig. 3. Generated views with respect to the missingness ratio r. In all the sub-figures, we show the missing data imputation from the first view to the second and vice versa. $\{G_i(Z_j)\}_{i,j\in\{1,2\}}$ is the view generated by the i-th generator from the j-th latent vector.

of accuracy when $r = 0.1$. However, the accuracy drops linearly when r increases which gives an insight on how the missing data could affect the MVE model. From the obtained results, we can also notice that, in the worst case (i.e $r = 0.9$), the accuracy is not lower than 0.4 while it exceeds 0.6 with most of the classic classifiers.

V. Comparison with SOA methods

In the MVE learning model, the AE, Fusion and GAN networks are paramount to encode the input images in view-specific latent space, common latent space and to impute the missing data, respectively. When exploring the state-of-the-art methods for multi-view representation using GANs, we found out that the GP-MVC model in [10] outperforms the rest of the techniques and is quite recent (2020). Therefore, we decide to compare our proposed model with it. We note that GP-MVC includes

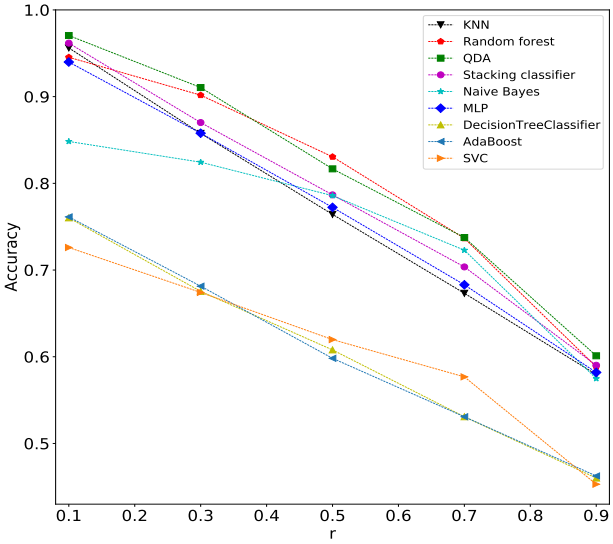


Fig. 4. Classification accuracy w.r.t. the missing data ratio r.

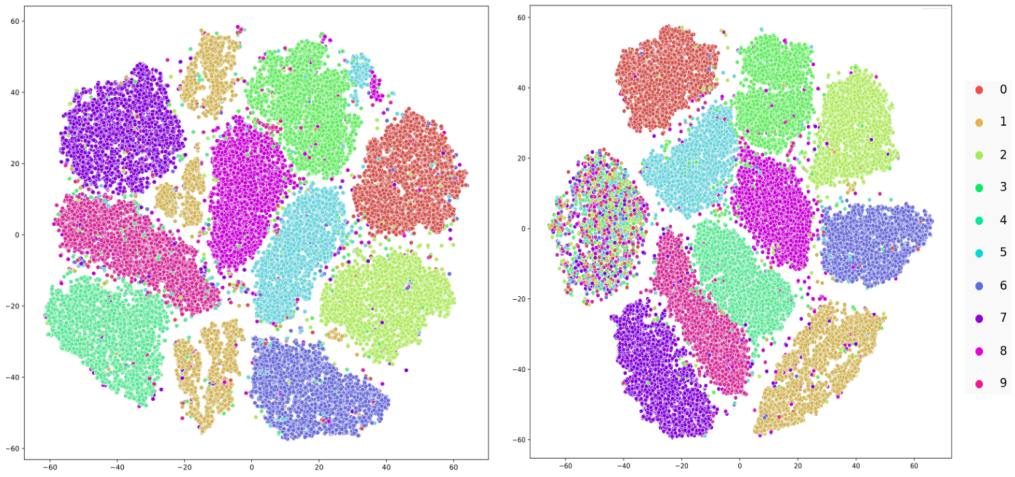
an extra clustering layer. Thus, we propose to investigate the impact of such component on the target embeddings.

First, we conduct the same experiments (i.e dataset, missingness ratios, classifiers) for both models and report the mean accuracy of the several classifiers in Table I. We notice that the classification accuracy of the two models differ by a few percentile points. However, our MIC model outperforms GPMVC for all missingness ratios except $r = 0.9$.

TABLE I
MEAN ACCURACY OF THE DIFFERENT CLASSIFIERS USED W.R.T r AND TRAINING PROCEDURE (BASED ON GP-MVC OR MIC).

r	0.1	0.3	0.5	0.7	0.9
GP-MVC	85.26	78.15	70.89	64	**56.87**
MIC	**87.43**	**80.59**	**73.13**	**65.5**	54.35

Secondly, we display the projection using T-SNE of the common latent space returned by our model MIC and GP-MVC [10] respectively (see Figure 5). MIC's clustering is not perfect and some points that belong to the same class (digit 1 in yellow) reside in three different regions in the common space. However, we do not identify any overlap between the clusters. On the other hand, the T-SNE projection of the latent vectors found by the GP-MVC model returns less noisy points due to its additional clustering loss. However, it generates an extra cluster which is nothing but a mixture of points belonging to different classes. From this experiment, we can confirm, once again, the superiority of our model in comparison to the state-of-the-art method GP-MVC [10]. We also conclude that the clustering layer is not necessary for an image classification task (especially if the missing data ratio is not high).



(a) MIC (b) GP-MVC

Fig. 5. T-SNE representation of the common latent space found by: (a) MIC, (b) GP-MVC.

VI. CONCLUSION

In this paper, we propose a framework for a multi-view image classification task. Indeed, we have suggested a GAN-based MVE architecture which takes as input multi-view images and embed them in a common latent space while imputing the missing view if any. The proposed framework is able to train the multi-view embedding model for extracting the visual high-level features, to generate the missing views and to generate a common representation that benefits from the complementary information between the views. As the used architecture is not rigid, the end-to-end pipeline is flexible and is adaptable to data variety (image resolution, number of views, etc.) and can even be extensible to other modalities such as text or audio. In future works, we also aim to integrate the feature extraction component within an autoML framework in order to automatically choose the best classifier and tune the hyperparameters.

REFERENCES

[1] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[2] Hasan Nasir Khan, Ahmad Shahid, Basit Raza, Amir Dar, and Hani Alquhayz, "Multi-view feature fusion based four views model for mammogram classification using convolutional neural network," *IEEE Access*, vol. PP, pp. 1–1, 11 2019.

[3] Lichen Wang, Zhengming Ding, Zhiqiang Tao, Yunyu Liu, and Yun Fu, "Generative multi-view human action recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6212–6221.

[4] Yue Bai, Zhiqiang Tao, Lichen Wang, Sheng Li, Yu Yin, and Yun Fu, "Collaborative attention mechanism for multi-view action recognition," *arXiv preprint arXiv:2009.06599*, 2020.

[5] Puneet Mishra, Ittai Herrmann, and Mariagiovanna Angileri, "Improved prediction of potassium and nitrogen in dried bell pepper leaves with visible and near-infrared spectroscopy utilising wavelength selection techniques," *Talanta*, vol. 225, pp. 121971, 2021.

[6] Z Yan, A J Mead, L Van Waerbeke, G Hinshaw, and I G McCarthy, "Galaxy cluster mass estimation with deep learning and hydrodynamical simulations," *Monthly Notices of the Royal Astronomical Society*, vol. 499, no. 3, pp. 3445–3458, Oct 2020.

[7] Y. Li, M. Yang, and Z. Zhang, "A survey of multi-view representation learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 10, pp. 1863–1883, 2019.

[8] "Vot 2020," https://www.votchallenge.net/vot2020/dataset.html.

[9] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[10] Qianqian Wang, Zhengming Ding, Zhiqiang Tao, Quanxue Gao, and Yun Fu, "Generative partial multi-view clustering," 2020.

[11] Chao Shang, Aaron Palmer, Jiangwen Sun, Ko-Shin Chen, Jin Lu, and Jinbo Bi, "Vigan: Missing view imputation with generative adversarial networks," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 766–775.

[12] Zhongkai Sun, Prathusha Sarma, William Sethares, and Yingyu Liang, "Learning relationships between text, audio, and video via deep canonical correlation for multimodal language analysis," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[13] N. Rai, S. Negi, S. Chaudhury, and O. Deshmukh, "Partial multi-view clustering using graph regularized nmf," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2016, pp. 2192–2197.

[14] Samuel K Ainsworth, Nicholas J Foti, and Emily B Fox, "Disentangled VAE representations for multi-aspect and missing data," *arXiv preprint arXiv:1806.09060*, 2018.

[15] Xiang Li, Chao Wang, Jiwei Tan, Xiaoyi Zeng, Dan Ou, Dan Ou, and Bo Zheng, "Adversarial multimodal representation learning for click-through rate prediction," *Proceedings of The Web Conference 2020*, Apr 2020.