SPARK CHALLENGE: MULTIMODAL CLASSIFIER FOR SPACE TARGET RECOGNITION

Ichraf Lahouli[⋆] *Mahmoud Jarraya*[⋆] *Gianmarco Aversano*[†]

[⋆] EURANOVA TN, Les berges du lac, Tunisia
[†] EURANOVA BE, Mont-Saint-Guibert, Belgium

ABSTRACT

In this paper, we propose a multi-modal framework to tackle the SPARK Challenge by classifying satellites using RGB and depth images. Our framework is mainly based on Auto-Encoders (AE)s to embed the two modalities in a common latent space in order to exploit redundant and complementary information between the two types of data.

Index Terms— Auto-encoders, Multi-modal, RESNET, Space Target Recognition

1. INTRODUCTION

Multi-modal images are quite common in visual systems and are often complementary as their are taken from different kinds of sensors or generated by different processing techniques. Thus, they are useful for various computer vision applications such as surveillance [1, 2], medical applications [3], and agriculture [4] but also for Space Situational Awareness (SSA) like in our case. The proposed SPARK challenge [5] provides a new unique space multi-modal annotated image dataset with a total of nearly 150k RGB images and the same number (i.e 150k) of depth images of 11 object classes (i.e 10 spacecraft and 1 class for space debris) [6]. This challenge offers the opportunity to work with multi-modal images generated under a realistic space simulation environment, with a large diversity in sensing conditions" making the space target recognition task very challenging.

The research field of Learning common (latent) representations of multi-modal is referred to in the literature as *multi-modal representation learning*. This goal is mainly achieved by either aligning or fusing modality-specific latent features. Another approach, called multi-modal embedding (MME), is the projection of these modalities in a common latent space. The resulting feature vectors can then be used to solve several ML tasks. Typical techniques include non-negative matrix factorization (NMF)-based methods [7] or kernel-based methods such as Deep Canonical Correlation Analysis (DCCA) [8]. However, they suffer from several limitations, mainly inefficiency to scale with large scale databases, complexity and wastefulness to compensate for missing modalities due to regularization and added constraints [9] which makes them less attractive. Recently, generative models such as GANs [10] or VAEs [11] have gained popularity in this field [9, 12] thanks to their effectiveness in narrowing the difference between the different modalities' distributions [13] and their ability to map any given distribution to a target one.

In a previous work [14], we have suggested an MME architecture, based on a mixture of AEs and GANs, which takes as input multi-modal images and embed them in a common latent space while imputing the missing modality if any. For the SPARK challenge, we propose an AE-based architecture only to extract the visual features from the RGB and depth images as no missing modalities exist. The modalities are embedded in a common latent space and trained in an unsupervised way. Then, we tried a panoply of classifiers that take as input the common features in order to recognize the satellites and distinguish between them and space debris.

2. PROPOSED METHODOLOGY

This section describes the MME model architecture, inspired by our previous work in [14] and how this model is used to generate the embeddings that are further exploited by the classification stage.

2.1. Notations

For the sake of generalization, we start by formalising the multi-modal input data which is usually represented by: $\mathbf{X} = \left\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(V)}\right\}$ where $\mathbf{X}^{(v)} = \left\{\mathbf{x}_1^{(v)}, \mathbf{x}_2^{(v)}, \cdots, \mathbf{x}_N^{(v)}\right\} \in \mathbf{R}^{N \times d_v} \forall v \in [1, V]$. V is the number of modalities, N is the number of samples, and d_v is the feature dimension of v-th modality. d is the feature dimension of the common latent space.
In the SPARK challenge, we will work with two modalities so $V = 2$, $\mathbf{X}^{(1)}$ represents the RGB modality and $\mathbf{X}^{(2)}$ corresponds to the depth modality. For the rest of the paper, we will denote them $\mathbf{X_{RGB}}$ and $\mathbf{X_{DEPTH}}$ respectively.
We denote by $\mathcal{Y}$ the labels space and each sample of the dataset (couple of RGB and depth images) is associated to $y_n \in \mathcal{Y}\ \forall n \in [1, N]$.

2.2. Architecture

The overall architecture consists of two components; the first is for learning a representative feature vector in a common latent space of the multi-modal input data through an unsupervised training and the second is used for the classification task purpose.

2.2.1. MME component

The given SPARK dataset is very challenging as the two modalities, RGB and depth, contain different characteristics and information. While $\mathbf{X_{DEPTH}}$ is mostly similar to a binary mask (in gray scale), $\mathbf{X_{RGB}}$ contains more information so the two modalities can not be treated similarly. In consequence, we make use of six different blocs in order to project the two modalities in a common latent space and profit from the multi-modal information:

ResNET18 $R18$: $R^{d_1} \rightarrow R^{d_{R18}}$. A pre-trained ResNET18 network [15] is used to extract high-level features of the RGB images. The network's output is a d_{R18}-dimensional feature vectors.

RGB Encoder E_1: $R^{d_{R18}} \rightarrow R^d$. This encoder projects the input RGB embedding $R18(X_{RGB})$ to a low-dimensional features subspace through a shallow fully connected neural network. It outputs a d-dimensional vector $Z^{(1)}$.

DEPTH Encoder E_2: $R^{d_2} \rightarrow R^d$. An encoder made up of stacked convolution-batch normalization-ReLu layers for extracting high-level features of its input DEPTH images. It outputs d-dimensional vector $Z^{(2)}$.

Fusion F: $R^{2d} \rightarrow R^d$. The intuition from the fusion network is to capture the shared semantics of the multi-modal data from the resulting V embedding vectors. It takes the output of the encoders and derives the fused target feature vector Z through a non-linear fully connected layer.

RGB Decoder D_1: $R^d \rightarrow R^{d_{R18}}$. This decoder projects the fused vector back to the embedding space. It has a mirrored architecture to E_1.

DEPTH Decoder D_2: $R^d \rightarrow R^{d_2}$. Similar to D_1, this decoder takes as input the fused vector and reconstructs the original DEPTH image inputted to E_2 through a stacked transposed-convolution layers.

2.2.2. Classification component

In order to exploit the common feature vectors outputted by F, we can directly either feed them directly to a classical algorithm (e.g K-nearest neighbors, random forest, support-vector machines) or add another neural networks bloc to the overall architecture in order to back-propagate the classification loss to fine-tune the earlier RGB and Depth Encoders.

Classification MLP head T: $R^d \rightarrow R^{11}$. Basically, it is a stack of non-linear fully-connected layers for extracting task-related features followed by an output layer (i.e classification layer C). It takes the multi-modal representation obtained by F and outputs a vector of probabilities of each class by C.

2.3. Objective Functions

We partition the problem in two sub-problems; the first one is to learn the multi-modal representation component in an unsupervised way and the second is to solve the task of classification. Thus, we optimize two objective functions separately.

Auto-Encoder loss: The purpose is that the multi-modal feature vector is representative enough and holds the cross-modality information. The reconstruction loss is defined as follow:

$$L_{rgb} = \|R18(X_{RGB}) - D_1(E_1(R18(X_{RGB})))\|_1 \quad (1)$$

with $R18(X_{RGB})$ being the embedding vector extracted by the Resnet18 network. Eq. 1 measures the distance between the original embeddings outputted by the ResNET18 network on the RGB images and the decoded embeddings outputted from the RGB decoder.

$$L_{depth} = \|X_{DEPTH} - D_2(E_2(X_{DEPTH}))\|_2^2 \quad (2)$$

Eq. 2 measures the reconstruction loss between the original depth images and the reconstructed depth images by the Depth Decoder.

The total reconstruction loss can be computed as

$$L_{rec} = \alpha L_{depth} + \beta . L_{rgb} \quad (3)$$

with α and β are weights that allow balancing the impact of each AE depending on the initial modality.

Classification task loss: We use the categorical cross-entropy loss to train our classification head.

$$L_{task} = -\log(\hat{y}_c) \quad (4)$$

Where $\hat{y}$ is the 1×11 output vector of the classification layer C containing the classes' probabilities, c is the ground truth class and $\hat{y}_c$ is the output energy for class c.

3. EXPERIMENTS AND RESULTS

3.1. Training Methodologies

In order to validate the aforementioned methodology, we first pre-train the AE networks in a totally unsupervised way by minimizing the reconstruction loss L_{rec}. Secondly, we experimented two supervised training methods (with/without the classification MLP head) to get the final classification results. Then, we do a comparison between the two methods based on a visualisation of the latent space and the performance on the validation set.

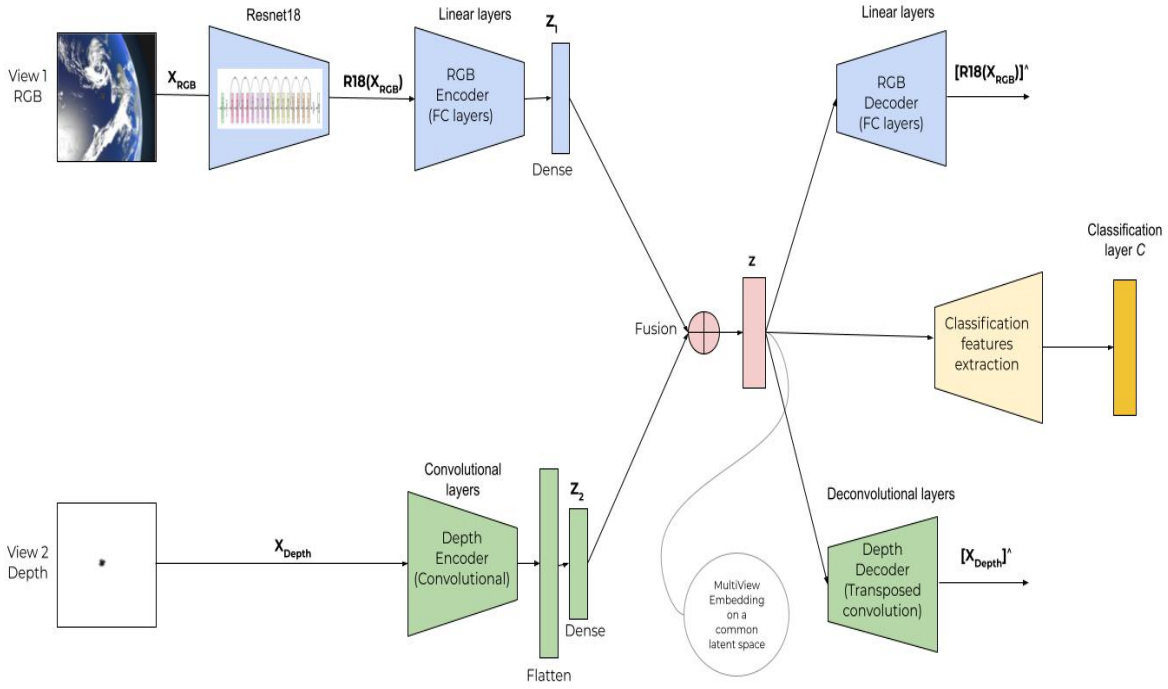


Fig. 1. Proposed multi-modal Classifier on the SPARK dataset

- **Method1**: The MME component (RGB and Depth Auto-encoders E_1 and E_2 and the fusion layer F) is trained in an unsupervised manner by minimizing the reconstruction loss L_{rec}. Then, we convert the input images to fused latent vectors and use these vectors as input features to some classical machine learning models (i.e K-nearest neighbors (KNN), random forest (RF, support-vector machines (SVM)).

- **Method2**: The method2 involves the classification MLP head mentioned at the 2.2 section. The training procedure becomes in three steps:

  1. The MME component is pre-trained in an unsupervised manner exactly as done in Method1.
  2. The Classification MLP head is trained in a supervised manner and then the classification task loss L_{task} is back-propagated to fine-tune the pre-trained networks E_1, E_2 and F. Thus, in this step, we perform a supervised end-to-end training of the MME component in combination with a classification MLP head.
  3. The E_1, E_2, and F networks are frozen, the classification MLP head is removed, and some classical ML models are trained on the resulting fused vectors such as in Method1.

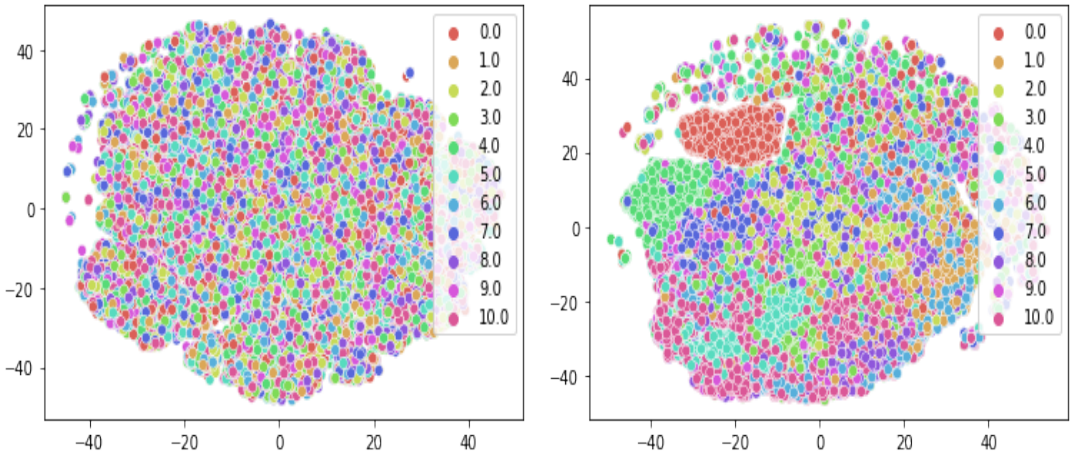
The goal of proposing these two training procedures is to check the impact of back-propagating the classification loss on the common latent space (i.e the fused vectors of the RGB and Depth modalities) on the downstream task performance.

3.2. Implementation Details

We set the latent dimension $d = 64$, $E1$ and D_1 to be a shallow linear layer each (mirrored to each other) and we made up E_2 and D_2 of 3 convolution and transposed convolution layers with ReLU activation. We train the AE networks on mini-batches for 20 epochs with $Adam$ optimizer with a learning rate $l_r = 0.0005$. The classification MLP head is made up of a hidden layer with 32-units, a ReLU activation, a Dropout layer, and the output layer C. We train the task head in addition to the fusion and encoders networks on mini-batches for 25 epochs with $Adam$ optimizer with a learning rate $l_r = 0.00001$.

3.3. Latent Space Visualization & Classification Results

Toward comparing between the two training procedures, we first visualize the common latent space using the t-SNE [16] projection. We report the results in Fig. 2. We notice that the fused vectors obtained by Method2 are grouped into clusters with respect to the class label unlike the latent space in Method1 which appears as a noise in terms of class clusters.



(a) Method1 (b) Method2

Fig. 2. T-SNE representation of the common latent space found after: (a) Method1 training procedure (AEs trained in fully unsupervised way without back-propagating the task loss), (b) Method2 training procedure (AEs pre-trained in unsupervised way then refined with the back-propagating the task loss through the classification MLP head).

Method Type	**Classification Model**	**Classification Accuracy (%)**
1	KNN	13.85
	SVM	18.95
	Random forest	18.87
2	KNN	39.19
	SVM	42.62
	Random forest	**43.56**

Table 1. Comparison of the classification performance between the two proposed training procedures and using different classification algorithms.

This can be explained by the fact that, in Method1, we learn features from the multi-modal data in unsupervised way (i.e based only on the reconstruction performance and without any information on the labels). However, in Method2, we applied the additional classification MLP head to refine the common features for the classification task.

Secondly, we report the classification results of each of the two training procedures in Table 1. The classical models such as KNN, SVM and Random forest, applied directly after the unsupervised training, reached low accuracy values. However adding the classification MLP head considerably improves the classification performance as it helps in getting better common representations by fine-tuning the early-stage AEs. For instance, the accuracy using the random forest algorithm jumps from less than 19% to more than 43%. The same observation is made with SVM and KNN algorithms (an improvement by a factor of 2 and 3 respectively).

4. CONCLUSION & DISCUSSIONS

For the SPARK challenge, we have suggested an AE-based MME (multi-modal embedding) architecture, consisting of encoders, fusion layer and decoders, which takes as input multi-modal images and embed them in a common latent space. The resulting common feature vectors are further exploited by a classification component in order to recognize the space target and solve the downstream supervised task.

We have tested two different training procedures: fully unsupervised auto-encoder training (to convert the multi-modal samples into common latent vectors) followed by a supervised training of some classical ML algorithms based on the resulting vectors and their corresponding labels (Method1); or unsupervised auto-encoder training, then fine-tuning the encoder and fusion networks with an MLP head in a supervised setting, and finally freezing the encoder and fusion networks and training classical ML methods (Method2). We observed that adding the extra classification MLP head improves the classification performance as it helps in getting better common representations by fine-tuning the early-stage AEs. This was also qualitatively visible on the common latent space projections.

In future works, we plan to combine this architecture with a scalable AutoML library, such as DeepHyper [17]. Such a framework will be able to, all at once, train the multi-modal embedding model, automatically choose the best classifier and tune all the hyper-parameters including the ones related to the neural networks that extract the visual high-level features.

5. REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] Mohamed Adel Musallam, Kassem Al Ismaeil, Oyebade Oyedotun, Marcos Damian Perez, Michel Poucet, and Djamila Aouada, "Spark: Spacecraft recognition leveraging knowledge of space environment," *arXiv preprint arXiv:2104.05978*, 2021.

[6] Mohamed Adel Musallam, Vincent Gaudilliere, Enjie Ghorbel, Kassem Al Ismaeil, Damian Perez Marcos, Michel Poucet, and Djamila Aouada, "Spacecraft recognition leveraging knowledge of space environment: Simulator, dataset, competition design, and analysis," in *2021 IEEE International Conference on Image Processing, Grand Challenges*, 2021.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Gianmarco Aversano, Mahmoud Jarraya, Maher Marwani, Ichraf Lahouli, and Sabri Skhiri, "Mic: Multi-view image classifier using generative adversarial networks for missing data imputation," in *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*. IEEE, 2021, pp. 283–288.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," 2015.

[16] L.J.P. van der Maaten and G.E. Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[17] P. Balaprakash, M. Salim, T. D. Uram, V. Vishwanath, and S. M. Wild, "Deephyper: Asynchronous hyperparameter search for deep neural networks," in *2018 IEEE 25th International Conference on High Performance Computing (HiPC)*, 2018.