Reproducing the results

The results can be reproduced by cloning the repository https://github.com/ciortanmadalina/FDNA and launching Experiments.ipynb

The code should be run in a python environment having installed the libraries listed in requirements.txt. This can be achieved by

Pip install -r requirements.txt

Experiments

The steps taken to implement the exercise have been described in Experiments.ipynb and on a high level they consist in:

- Loading the input data into a single dataframe containing denormalized properties for syndromes/embeddings etc
- Splitting it into train-test (e.g. 80/20 %)
- Training a clustering algorithm (e.g. KMeans) on the train set and predicting the classes on the test set
- The evaluation requires matching the predicted clustering labels with the ground truth. For simplicity a greedy approach matching the most frequent ground truth class to each predicted cluster has been used. However this technical choice can be improved using for example dedicated label alignment algorithms
- The evaluation uses a one-vs rest AUC metric which requires binarizing the predicted/ground truth labels and computing for each class the AUC

d) Write a short paragraph with your conclusions (can we use this method to distinguish between those syndromes?)

The selected method manages to correctly distinguish 9/10 syndromes with AUCs above 0.74. However, syndrom 4 is highly confused with syndrom 0 and syndrom 6.
There results could be improved by:

- Exploring other clustering algorithms especially the ones that don't require a fixed number of clusters to be provided as input and offer the flexibility to discover the clusters based on density (e.g. DBSCAN/HDBSCAN) or distance (e.g. hierarchical clustering) thresholds

- Improving the embedding space by making it smoother and optimizing for internal quality (e.g. using contrastive self-supervised methods)
- Handling the class imbalance (e.g. Smote can oversample underrepresented samples)

a) How would you create these embeddings? How would you optimize and measure the accuracy of the whole solution?

The used embeddings can be generated either as one of the final layers of a classifier trained to distinguish between syndromes. However, this baseline approach does not guarantee that the embedding space has properties facilitating future clustering (e.g. smoothness, interpolability).

One improved approach would be to use unsupervised or self-supervised techniques. The supervised contrastive learning (https://arxiv.org/abs/2004.11362), is one approach which is expected to create an embedding space with an improved internal clustering quality.

b) Besides the patient photo, we optionally allow geneticists to select phenotypic abnormalities that are present/absent (e.g. long face, tall stature, etc.). How would you design a research project on leveraging these features to improve the syndrome classification results?

Using multi-modal models leveraging a fusion gate tolerant to missing data (e.g. using operations such as sum/product of experts for combining the different modalities instead of concatenation) is one strategy allowing us to combine a number of different inputs without enforcing their presence. Methods such as https://ieeexplore.ieee.org/document/9429478 (summarized in Fig 1) also provide generation for missing views, an application which brings an added value in itself.

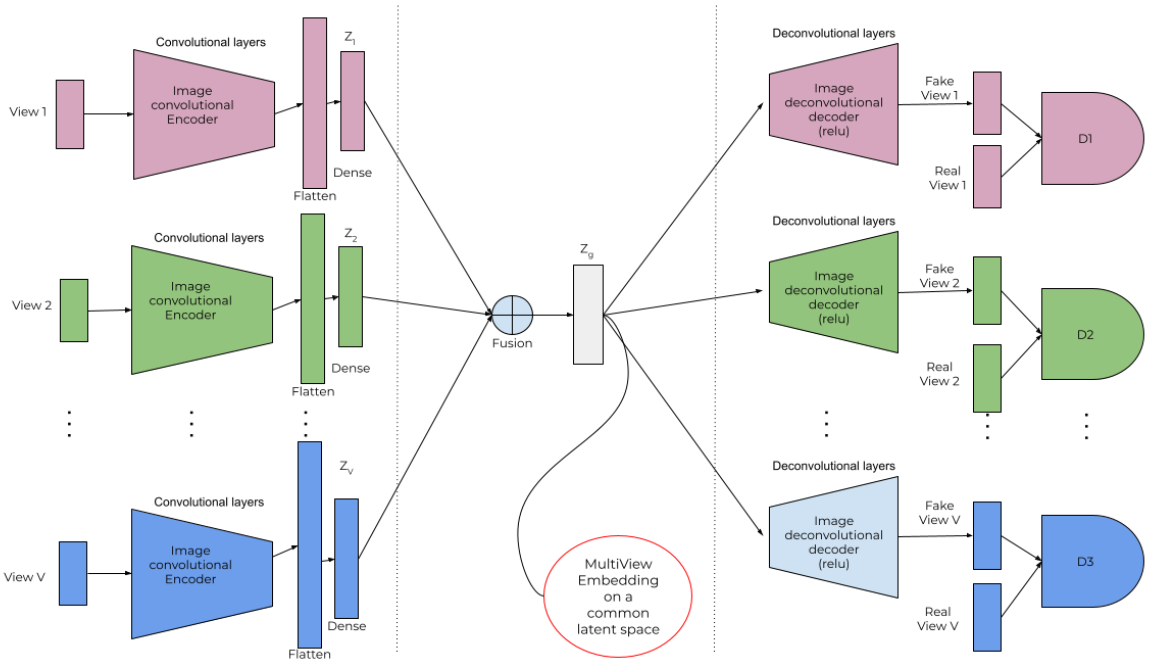

Fig 1. Architecture of a multi view embedding model combining multiple input sources while learning to reconstruct the input and generate missing data (i.e. imputation).