Assignment 3: Dimensionality Reduction and Visualization of Deep Learning Architectures

DSC 415: Data Analysis and Visualization

Nupur Upadhyay
IMS22191
Indian Institute of Science Education and Research, Thiruvananthapuram
Kerala, India
nupur22@iisertvm.ac.in

***Abstract*—This report presents the application of dimensionality reduction on the dataset used in Assignment 2 - Cancer Prediction Dataset by Rabie El Kharoua (Kaggle). The dimensionality reduction techniques of PCA and t-SNE with two and three components were applied on the dataset and visualised. The clustering performance of both techniques was compared. The second part of this report involves visualization of deep learning architectures like LeNet, AlexNet and more using the VisualKeras library.**

***Index Terms*—dataset, visualization, Dimensionality Reduction, Deep Learning, CNN**

I. Introduction

High-dimensional datasets are common in modern data science and biomedical research, often containing a large number of features that make analysis and visualization challenging. Dimensionality reduction techniques help address this issue by transforming high-dimensional data into lower-dimensional representations while preserving meaningful structure and relationships within the data.

This report focuses on the application of dimensionality reduction methods—Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE)—on the Cancer Prediction Dataset by Rabie El Kharoua from Kaggle. Both techniques were implemented with two and three components to explore and visualize the data's structure. The clustering capabilities of PCA and t-SNE were then compared to evaluate their effectiveness in distinguishing data patterns.

In the second part of this report, several deep learning architectures, including LeNet, AlexNet, and others, are visualized using the VisualKeras library in Python. These visualizations provide insights into the hierarchical composition and depth of each neural network model, aiding in the understanding of their architectural design and complexity.

Overall, this report combines data visualization through dimensionality reduction with neural network visualization, offering an integrated view of both data representation and model structure in computational analysis.

II. Dimensionality Reduction

The dataset consists of eight features along with a target variable (cancer diagnosis). Analysis of the feature pair plots shows no linear trends, suggesting that the interactions among features are complex and predominantly non-linear rather than straightforwardly linear (Figure 1).

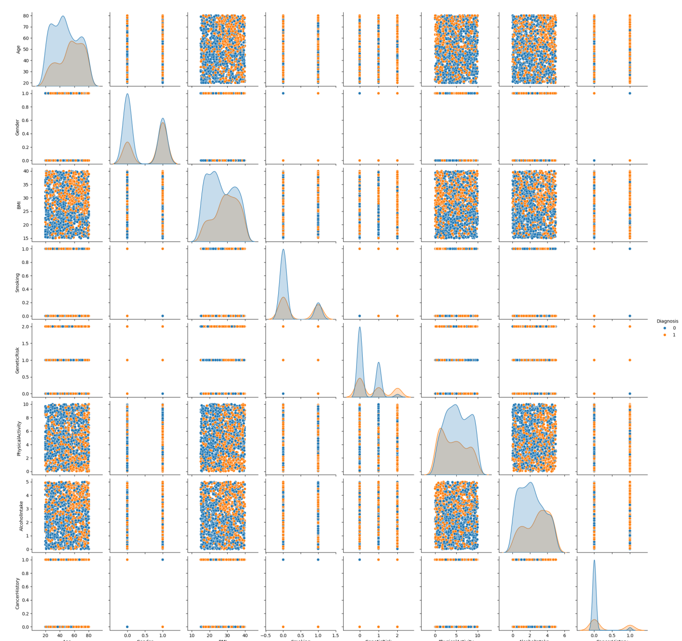

Fig. 1. Pair plots of all features

The two dimensionality reduction techniques, Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE), were applied to the dataset using the Scikit-learn library in Python. Before performing dimensionality reduction, the dataset was standardized using Scikit-learn's StandardScaler to ensure equal contribution from all features.. Standardization is a crucial preprocessing step, as PCA and t-SNE are sensitive to the scale of the data; unscaled features with larger numerical ranges can disproportionately influence the results.

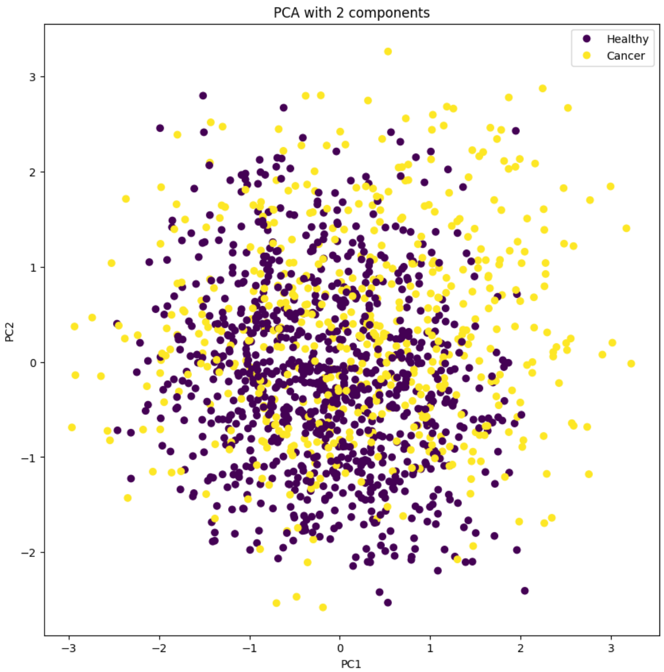

Fig. 2. PCA with 2 components

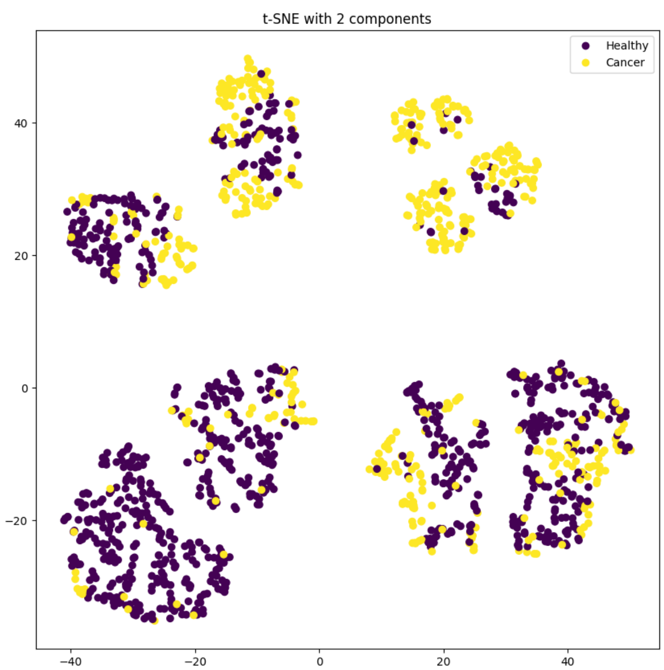

Fig. 3. t-SNE with 2 components

After scaling, both PCA and t-SNE were fitted and transformed on the dataset to obtain lower-dimensional representations. The PCA method was used to project the data onto two principal components that capture the maximum variance within the dataset, thereby providing a linear transformation of the original features. In contrast, t-SNE, a nonlinear technique, was employed to capture more complex, nonlinear relationships among data points by preserving their local similarities in the reduced space.

The resulting two-component PCA and t-SNE visualizations are presented in Figures 2 and 3, respectively. The PCA plot shows a relatively random distribution of data points, indicating that the linear transformation used by PCA is not sufficient to capture the complex structure of the dataset. This outcome is expected because the dataset contains non-linear relationships among features, which PCA, being a linear dimensionality reduction technique, cannot adequately represent.

In contrast, the t-SNE plot effectively captures these non-linear relationships, revealing a clear separation of data points into eight distinct clusters. revealing a clear separation of data points into several clusters, reflecting similarities among samples based on their feature patterns, demonstrating t-SNE's strength in preserving local similarities and uncovering the intrinsic structure of the data. This comparison highlights the advantages of non-linear dimensionality reduction techniques like t-SNE for datasets where feature interactions are complex and not linearly separable. The clustering observed in t-SNE provides valuable insights into the underlying patterns of the dataset, which might otherwise remain hidden in a linear projection such as PCA.

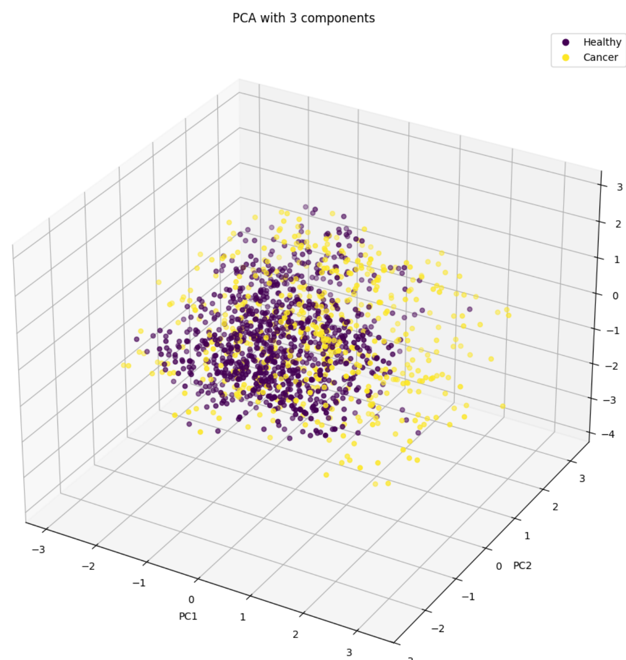

Fig. 4. PCA with 3 components

Similar trends were observed when the dataset was projected onto three components using PCA and t-SNE (Figures 4 and 5, respectively). The three-component PCA continued to show a diffuse distribution of points without clear clustering, indicating its inability to capture the non-linear structure of the data. In contrast, three-component t-SNE effectively separated the data into eight distinct clusters, consistent with the two-component results, further demonstrating its robustness in representing complex, non-linear relationships among the features of the dataset.

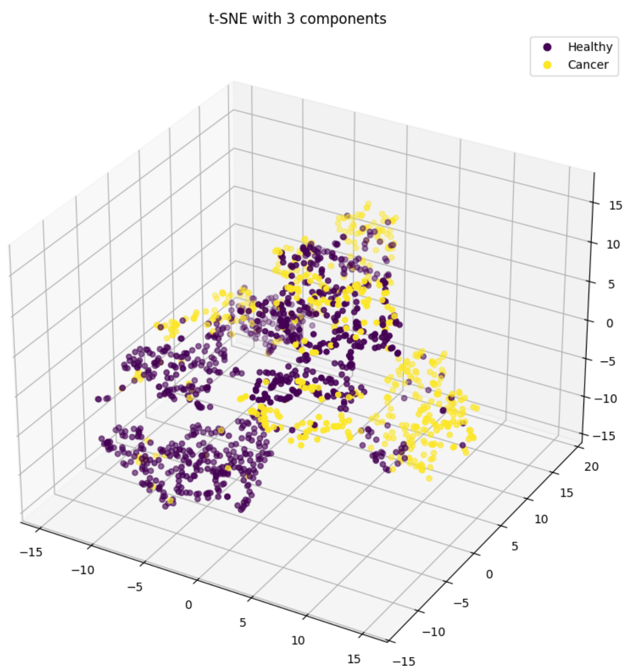

Fig. 5. t-SNE with 3 components

III. Visualization of Deep Learning Architectures

To gain a better structural understanding of various convolutional neural network (CNN) architectures, the models LeNet, AlexNet, VGG, ZFNet, and SegNet were visualized using the VisualKeras library in Python. VisualKeras provides a clear layered view of neural network architectures, displaying each layer as a distinct block with proportional depth and annotated legends indicating layer types.

The visualizations highlight how each model differs in complexity and design philosophy. LeNet, one of the earliest CNNs, has a simple and shallow architecture, demonstrating the foundational concept of convolution and pooling layers for image recognition. AlexNet expands upon this with deeper layers and a greater number of filters, marking a significant leap in performance for large-scale image classification. VGG employs a uniform architecture with multiple stacked 3×3 convolutional layers, emphasizing depth and simplicity. ZFNet refines AlexNet by optimizing filter sizes and stride, resulting in better feature extraction and interpretability. Finally, SegNet, designed for semantic segmentation tasks, features an encoder–decoder structure that maps pixel-level features back to image space, allowing precise spatial predictions.

Overall, the visualizations provide a comparative perspective of the evolution of CNN architectures — from early compact networks like LeNet to deeper and more application-specific models like SegNet — enabling a clearer understanding of how architectural choices influence model performance and complexity.

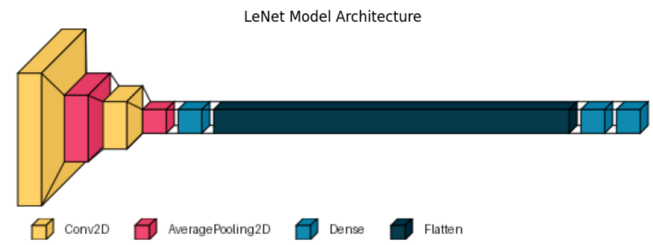

Fig. 6. LeNet Architecture

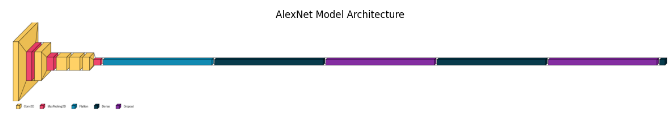

Fig. 7. AlexNet Architecture

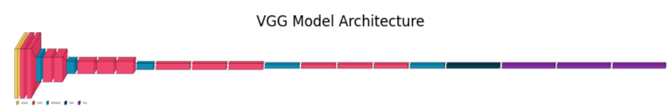

Fig. 8. VGG Architecture

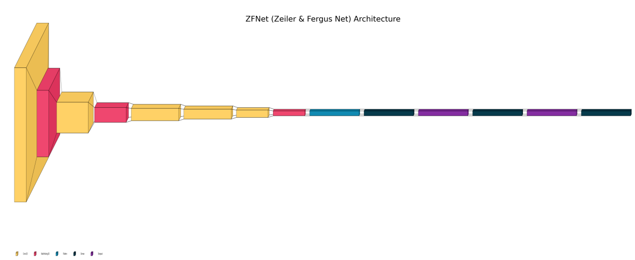

Fig. 9. ZFNet Architecture

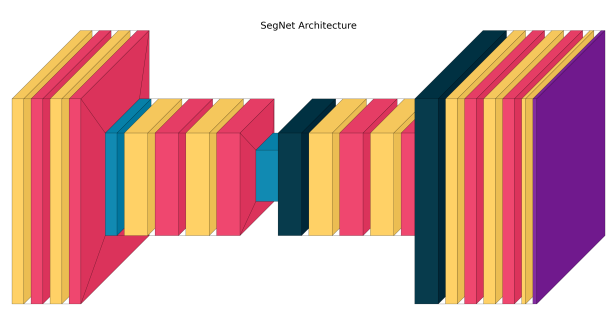

Fig. 10. SegNet Architecture