Concept Clusters

Project Report

Dominik Wolf

February 2023

Supervisor: Assistant Prof. Dr.techn. Manuela Waldner, MSc

1 Introduction

The Motivation for this work is to explore methods for visualizing high-dimensional datasets, with a specific focus on understanding the inner workings of Deep Neural Networks (DNNs). DNNs are a type of Artificial Neural Network (ANN) that consist of an input layer, an output layer, and multiple hidden layers. They are able to make predictions or decisions based on unseen observations, but the way that they arrive at these decisions can often be opaque, or difficult to interpret. By exploring the similarities of feature vectors in the latent space created by the network, it is possible to gain insight into how the network organizes and classifies observations with similar properties that are also interpretable to the human eye.

A common approach to visualize the clusters of observations in high-dimensional feature spaces is dimensionality reduction (DR). By reducing the number of dimensions to a lower number, typically two or three, the observations can be plotted on a surface and can be visually inspected and interpreted. Although this technique already can reveal patterns and trends, much information on the neighborhood relationships between individual observations in the high-dimensional space can be lost. Furthermore, DR introduces error phenomena like *missing neighbor* and *false neighbor* [1], both related to the issue of preserving the intrinsic structure of the data: Missing neighbors occur when important information is discarded, causing observations that should be close to each other in the original space to be separated in the reduced space. This results in missing or incomplete clusters in the reduced space. False neighbors occur when artificial relationships between observations are introduced, causing observations that are not similar in the original space to be grouped together in the reduced space. This may result in the formation of false clusters in the reduced space.

Previous work includes *Concept Splatters* by Grossmann et al. [2], which served as the starting point for this project. *Concept Splatters* allows users to explore similarities within datasets in a two-dimensional visualization space. Grossmann et al. proposed a method to visualize large latent spaces by displaying a select number of representative items for each cluster. However, this approach suffered from typical DR issues because the features were reduced to two dimensions utilizing UMAP [3] and clusters were computed afterwards in this two-dimensional visualization space. This work seeks to address this limitation by performing clustering in high-dimensional space before visualizing the clusters using representative samples and their connections in the original feature space.

In related work for clustering data in a high dimensional feature space, Ventocilla et al. [4] present two methods to improve the Growing Neural Gas (GNG) algorithm. GNG is a topology learning algorithm which models a data space using a Hebbian learning rule, originally proposed by Fritzke [5].

The goal of this work is to develop a prototype that enables interactive exploration of data clusterings in high-dimensional feature spaces, instead of the traditional two-dimensional visual representation. Since this project pursued a similar goal as *Concept Splatters*, but clustering is carried out in the high-dimensional feature space, the project is subsequently referred to as *Concept Clusters*.

2 Implementation

The following sections explain various aspects of the components of *Concept Clusters* which were incorporated into the final interactive visualization.

2.1 Clustering

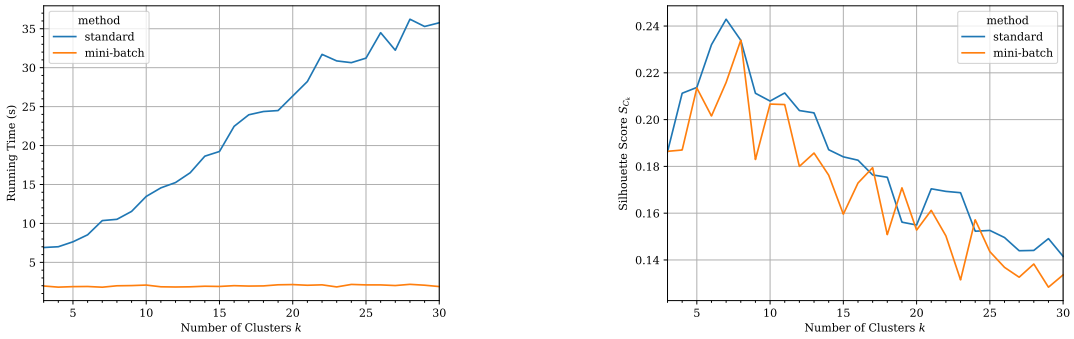
For the development of the first prototype, the Fashion-MNIST dataset [6] was utilized. This dataset includes 70,000 grayscale images, each with a resolution of 28x28 pixels and divided into ten distinct classes.

Various clustering algorithms were evaluated for their suitability for clustering data sets in a high-dimensional feature space. The algorithms were tested for both runtime performance and their mean silhouette coefficient, a measure of the similarity of an observation to its own cluster compared to other clusters. The silhouette coefficient ranges from -1 to 1, where a value of 1 indicates that the data point is highly similar to its own cluster and dissimilar to neighboring clusters, while a value of -1 indicates the opposite. A value of 0 means that the data point is close to the boundary between clusters. The clustering algorithms used were K-Means, DBSCAN and agglomerative clustering, all of which are available in the scikit-learn Python library. Additionally, a mini-batch variant of K-Means was also tested which uses a randomly selected subset of the input data, thus providing faster computation times.

The results showed that K-Means, especially in the mini-batch variant, produced the fastest results by a significant margin, making it the only algorithm considered for use in an interactive visualization. The mini-batch variant of K-Means consistently had a runtime of around two seconds for clustering the whole Fashion-MNIST dataset ($n = 70,000$), regardless of the number of clusters to be formed (k). In contrast, the standard variant's runtime increased as k increased, but still remained below 30 seconds for up to $k = 21$ clusters. On the same machine, DBSCAN took approximately 50 seconds to complete a clustering, while agglomerative clustering took over six minutes. Furthermore, the evaluation of the clusters showed that K-Means produced good results in terms of the mean silhouette coefficient. Given the fast runtime and good clustering performance, no further steps were taken to improve the clustering using other methods such as compressing the feature vectors with autoencoders or using a random subsample.

The mini-batch variant of the K-Means algorithm achieved the highest mean silhouette coefficient with $k = 8$ clusters. This value is close to the number of ground-truth classes in the Fashion-MNIST dataset of ten classes, therefore $k = 10$ was selected for use in the visualization process. Running times and mean silhouette coefficients for clustering the Fashion-MNIST dataset with different numbers of clusters are depicted in Figure 1.

For datasets with a small number of classes, setting the value of clusters k based on the



(a) Running times

(b) Mean silhouette coefficients

Figure 1: Running times and mean silhouette coefficients for K-Means clustering of the whole Fashion-MNIST dataset ($n = 70,000$).

number of ground-truth classes is feasible. However, this approach becomes impractical for datasets with a large number of classes. A more efficient approach would be to determine an optimal value or a range of values for k in a preprocessing step or enable setting it through user input, allowing for greater flexibility.

2.2 Interface Areas

The visualization is divided into three distinct areas, each serving a specific purpose.

In the visualization area (see Figure 2, A), clustering results are rendered as an undirected graph, where each cluster is represented as a node. The visual representation of a node is chosen based on the image of an observation in the cluster, that is closest to the cluster center. The relative size of a node in comparison to the other nodes in the graph is determined by the number of observations inside the cluster, where larger clusters are represented by larger nodes. The edges of the graph and the proximity of nodes to each other are determined by the distance between the cluster centers. As the distance between two cluster centers decrease, the edges connecting the corresponding nodes becomes more visible and the nodes are placed closer together in the visualization.

The concept space area (see Figure 2, B) is used to visualize the hierarchical relationship of labels in the data set using an icicle plot. The higher up on the icicle plot, the more general the labels are, while the labels become more specific as the plot progresses downward. The root and the inner nodes of the tree are "virtual" labels used solely for the hierarchy, while the leaves of the tree correspond to the ground-truth labels of the observations in the clustering graph.

A detail area (see Figure 2, C) allows for switching between the available data sets, resetting the current display each time. If a cluster is selected in the main view, various information about the selected cluster is displayed in this area. This includes the cluster's internal ID, the number of observations contained within the cluster, and examples of the images in the cluster. For each contained ground-truth label, up to ten random images from the corresponding observations are displayed, along with their relative and absolute representation within the selected cluster. In this area, it is also possible to create a sub-cluster from the selected cluster, which is further described in subsection 2.3.

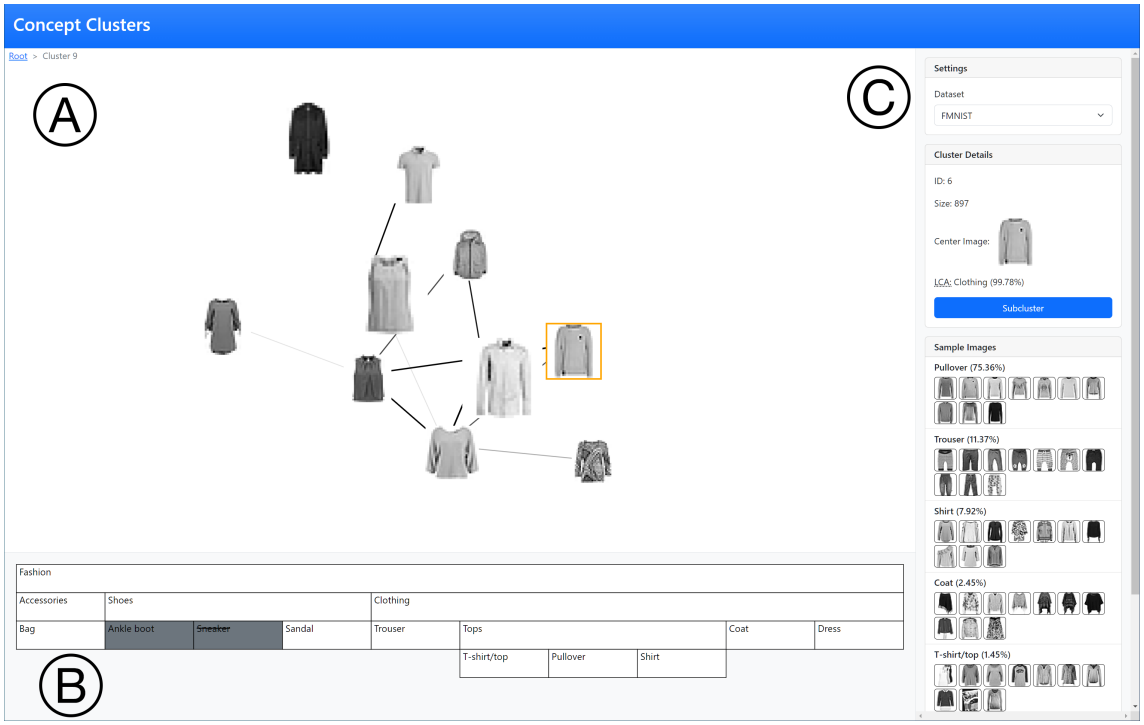


Figure 2: User interface of *Concept Clusters*. The three main areas are the visualization area (A), the concept space area (B), and the detail area (C).

2.3 Interaction

The three areas described above are strongly connected in terms of interaction.

2.3.1 Cluster Selection and Sub-Clustering

The visualization area displays the cluster graph, allowing users to select a specific cluster for further inspection in the detail area. Here, a sub-clustering can be performed by clicking on a designated button. This will re-cluster all elements of the selected cluster. To aid in navigation, a breadcrumb is also created for each sub-clustering and displayed in the upper area of the visualization. This allows users to easily return to previous cluster graphs and track the sub-clustering history.

In addition, the icicle plot in the concept space area also allows users to perform sub-clustering on the currently displayed cluster graph by clicking on a label. This filters the observations of all nodes in the graph by the selected label and its corresponding sub-labels in the hierarchy and clusters the data again. The visualization is updated to reflect the new cluster graph. Similar to sub-clustering in the detail area, a breadcrumb is also created for navigation purposes.

2.3.2 Hover Effects

To provide additional context and make related data more visible, various hover effects have been implemented that highlight certain elements in the visualization. When the mouse pointer is moved over a label in the icicle plot in the concept space area, all nodes in the cluster graph that contain at least one element of the respective label or one of its sub-labels in the hierarchy will be highlighted (see Figure 3). To further clarify the relationship, if an image of the corresponding label is not currently being used as the

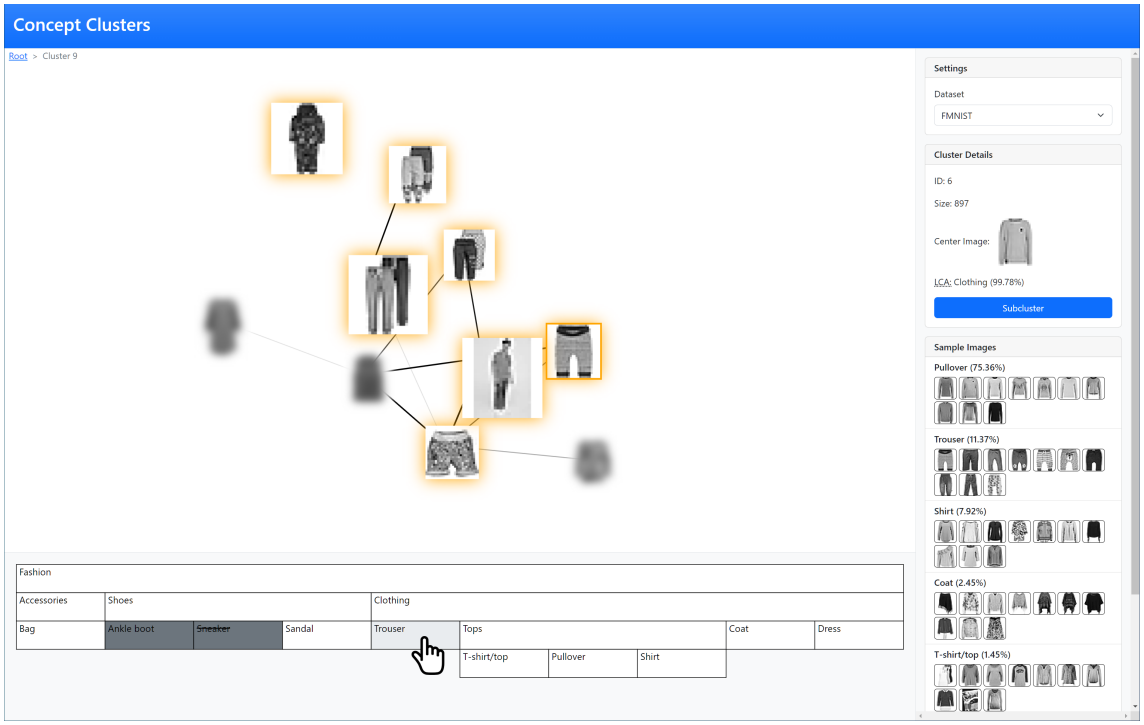


Figure 3: Hovering over a label in the concept space highlights matching elements within the other clusters, while blurring out clusters without any matching elements.

image of the cluster node, it will be temporarily replaced with the image closest to the respective cluster center for the duration of the hover effect.

A similar interaction option is also available in the detail area. If a cluster is selected and its contained sample images are displayed, hovering over a contained label will also highlight all clusters that contain observations of the respective label.

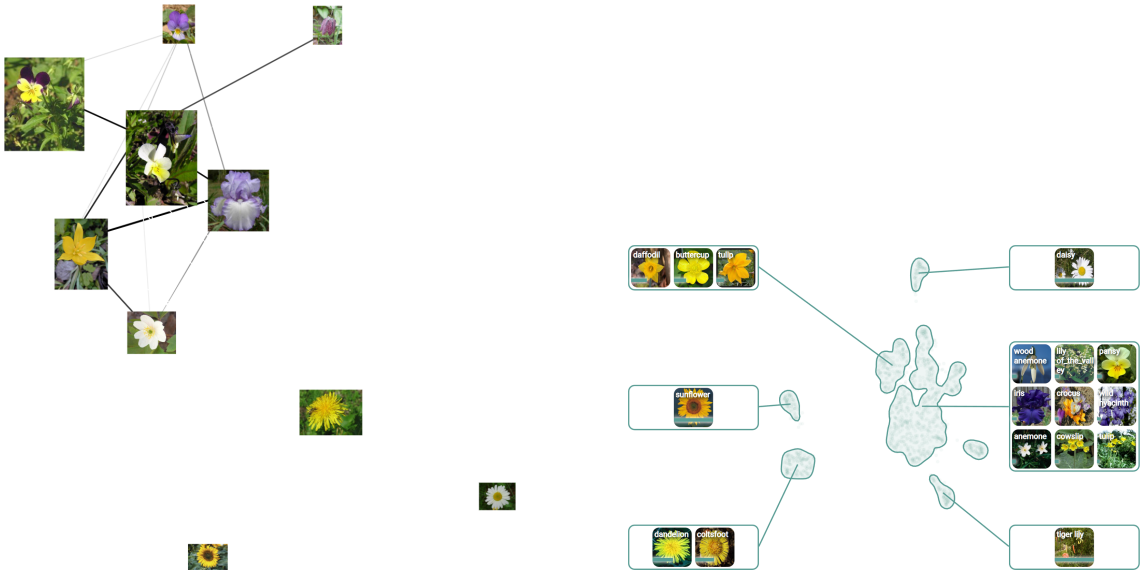
2.4 Technology Stack

The application is designed with a client-server architecture, where the server is implemented in Python and uses Flask as a web server. Data is transferred to the client as JSON (JavaScript Object Notation) objects. The NetworkX library was used for implementing the internal representation of the clustering graphs and label hierarchy trees. Scientific computing libraries such as NumPy, SciPy, and scikit-learn were used for the clustering calculations, including the computation of the silhouette coefficients and pairwise distances between observations, as well as basic array creation and manipulation.

On the client-side, the application is implemented in TypeScript, a typed superset of JavaScript. React was used as the UI framework and Bootstrap as the CSS framework. The clustering graph and icicle plot are rendered using D3. The state-managing library zustand was used to manage the application state.

2.5 Additional Datasets

The application is designed to be extensible, allowing for easy integration and analysis of additional datasets. Along with the Fashion-MNIST dataset, the *17 Category Flower Dataset* [7] has been included into the application, which comprises 1,360 images of various flowers across 17 ground-truth classes, with 80 images for each class. At $k = 10$,



(a) Clustering in *Concept Clusters*. (b) Clustering in *Concept Splatters*.

Figure 4: Initial clustering of the *17 Category Flower Dataset* in *Concept Clusters* and *Concept Splatters*.

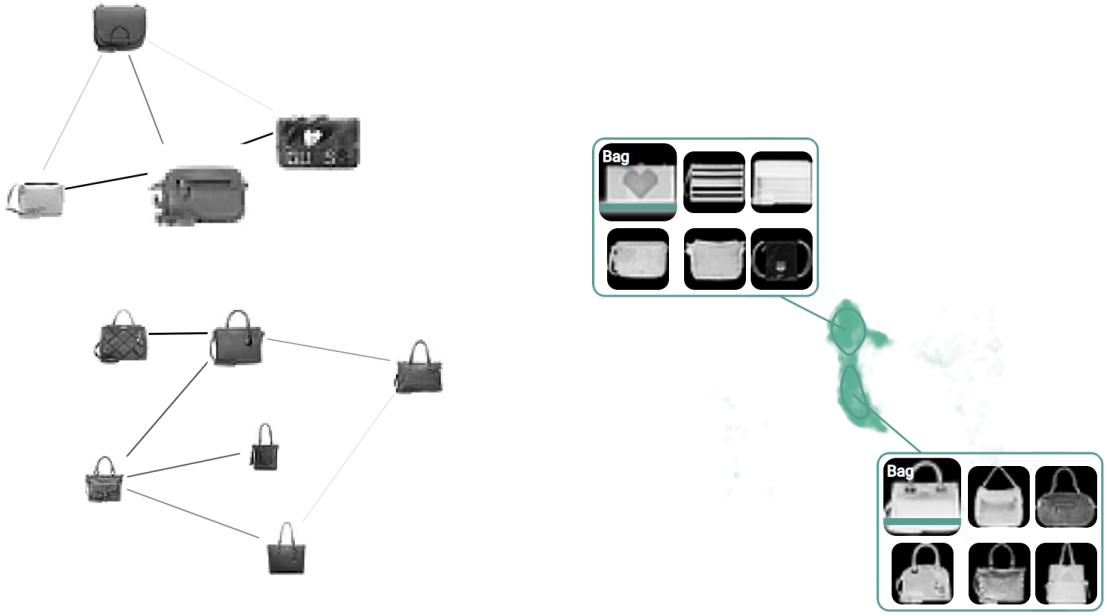
the silhouette coefficient of this dataset is approximately 0.1, indicating that its clustering performance in high-dimensional feature space is inferior to that of the Fashion-MNIST dataset. Since the *17 Category Flower Dataset* is also available in *Concept Splatters*, it makes sense to compare the initial clustering results of both applications. As depicted in Figure 4, both applications produce similar clusters, with detached clusters containing significant proportions of ground-truth labels such as "Sunflower," "Dandelion" and "Coltsfoot," and "Daisy." Meanwhile, other classes appear to be closer to each other in the feature space, which is visualized in *Concept Clusters* through the edges connecting individual clusters. These similarities between the two applications suggest that the implemented approach is effective.

3 Results

This section presents some insights that were uncovered via the visualization prototype, focusing on different clusterings.

3.1 Bag Clustering

By only clustering elements within the "Bag" label, K-Means effectively separates the images into two distinct groups of clusters: bags without handles or with minimal handles, and bags with prominent handles. The result of this clustering is depicted in Figure 5a. This suggests that the model has effectively learned the presence and size of a handle as a distinct characteristic in the high-dimensional feature space. Furthermore, if the clustering algorithm is applied to the whole dataset, it again becomes evident that the bags tend to cluster into two main groups. As previously observed, one cluster consists of bags with handles, while the other cluster is comprised of bags without handles. As depicted in Figure 5b, *Concept Splatters* likewise separates bags well from the rest of the dataset.



(a) *Concept Clusters*: Bags are clustered into two groups: bags with and without handles.

(b) *Concept Splatters*: Bags are well separated from the rest of the dataset.

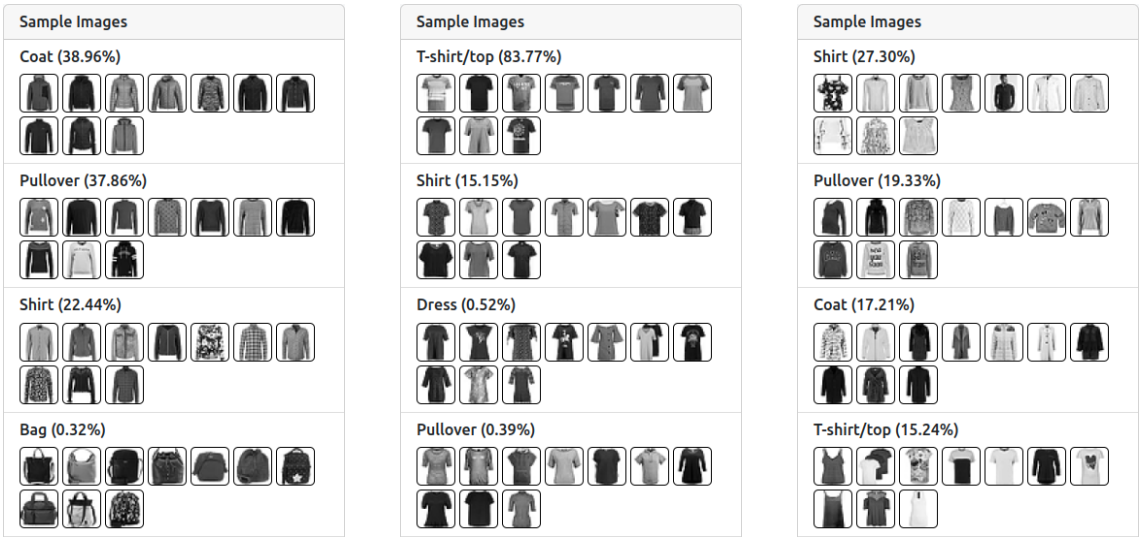
Figure 5: Clustering of bags in *Concept Clusters* and *Concept Splatters*.

3.2 Sleeve Clustering

When clustering the whole Fashion-MNIST dataset, it becomes apparent that the length of the sleeves of clothing items is a discriminative feature regardless of their actual ground truth labels. While the label "Shirt" encompasses both long and short sleeves, images of short-sleeved shirts tend to cluster with T-shirts that have similar sleeves, while long-sleeved shirts are often grouped with sweaters and jackets that feature long sleeves. Results of these clusterings are depicted in Figure 6.

4 Conclusion

In conclusion, the visualization of the results of the K-Means algorithm can reveal interesting findings about the underlying structure of the data. This prototype offers similar insights as *Concept Splatters*, but with improved responsiveness. The 2D representation in *Concept Splatters* often results in excessive overplotting, while the undirected graph visualization utilized by this prototype eliminates this issue. The K-Means algorithm is heavily dependent on its hyperparameter k, which determines the number of clusters that will be created in the data. The value affects the shape and size of the clusters, the distances between the centroids and the data points, and the overall performance of the algorithm. If k is set too low, the clusters may not accurately reflect the underlying structure of the data, while if set too high, the algorithm may become too complex and produce clusters that are overly specific. For future improvements, k could be configurable via the user interface. This would allow users to experiment with different clusterings and help finding an optimal value that best suits the data. Additionally, allowing different values of k for each sub-clustering would provide further flexibility and allow for a more nuanced



(a) Cluster with $n = 13,026$ samples.

(b) Cluster with $n = 5,379$ samples.

(c) Cluster with $n = 10,455$ samples.

Figure 6: Various sleeve lengths of Fashion-MNIST items are clustered together ($k = 10$).

analysis.

The source code for this implementation is available at `https://gitlab.cg.tuwien.ac.at/waldner/concept-clusters`.

References

[1] Luis Gustavo Nonato and Michaël Aupetit. "Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment". In: *IEEE Transactions on Visualization and Computer Graphics* 25.8 (2019), pp. 2650–2673. DOI: 10.1109/TVCG.2018.2846735.

[2] Nicolas Grossmann, Eduard Gröller, and Manuela Waldner. "Concept Splatters: Exploration of latent spaces based on human interpretable concepts". In: *Computers & Graphics* 105 (2022), pp. 73–84. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2022.04.013. URL: https://www.sciencedirect.com/science/article/pii/S0097849322000656.

[3] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2018. DOI: 10.48550/ARXIV.1802.03426. URL: https://arxiv.org/abs/1802.03426.

[4] Elio Ventocilla et al. "Scaling the Growing Neural Gas for Visual Cluster Analysis". In: *Big Data Research* 26 (2021), p. 100254. ISSN: 2214-5796. DOI: https://doi.org/10.1016/j.bdr.2021.100254. URL: https://www.sciencedirect.com/science/article/pii/S221457962100071X.

[5] Bernd Fritzke. "A Growing Neural Gas Network Learns Topologies". In: *Advances in Neural Information Processing Systems*. Ed. by G. Tesauro, D. Touretzky, and T. Leen. Vol. 7. MIT Press, 1994. URL: https://proceedings.neurips.cc/paper/1994/file/d56b9fc4b0f1be8871f5e1c40c0067e7-Paper.pdf.

[6] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. Aug. 28, 2017. arXiv: `cs.LG/1708.07747 [cs.LG]`.

[7] Maria-Elena Nilsback and Andrew Zisserman. "A Visual Vocabulary for Flower Classification". In: *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2006, pp. 1447–1454.