Visual Search Using Deep Embeddings

1st Lipi Chakraborty 2nd Siana Dicheva 3rd Quincy Sproul 4th Chenghao Zhang 5th Yufeng Yang

6th Weiguang Wang

***Abstract*—Visual search has been a known technique, used in search engines in the past few decades with GOOGLE and Pinterest being some of the most famous ones. From popular games such as Where's Wally [1] to medical experiments in brain detection in electroencephalography (EEG) and functional magnetic resonance imaging (fMRI), the methodology is useful in feature extraction and analysis of information.**

Using visual search to promote similar clothes has been popular in the fashion industry. However, traditional visual search methods of feature extraction and matching become less effective because clothing items are often photographed from different angles. The purpose of our project is to apply modern deep-embedding machine-learning techniques to optimize visual searching.

***Index Terms*—deep embedding, visual search, model comparison, CNN, VGG, DEC, ResNet, contrastive learning, transfer learning**

I. Introduction

Visual search engines allow users to search for information via images rather than text or phrases. On input, the engine will analyze images, identify key features, and then use the information to find items with similar visual features. This has proven useful in a variety of industries as described in the section II and can be helpful in a multitude of tasks.

In the fashion industry, clothing items are often photographed from different angles, constituting visual searching a difficult task. Images can serve as a mediator between modern technology and culture, thus it remains that visual search has high importance in technology and further industries. While there have been significant developments in search engines, whether it be through personalizations or natural language processing, searching images remains less prominent comparison [2]

With studies showing that 90% of information transmitted to the human is visual, or that the human brain can identify images seen for as little as 13 milliseconds, it stands to reason why 45% of retailers in the UK now conduct a visual search [3].

Identify applicable funding agency here. If none, delete this.

Fig. 1: Top from different angles

With the main providers of visual search engines being Pinterest, Google and Microsoft, and a variety of retailers such as Argos having built proprietary visual search tools, this paper aims to provide reasoning into implementations of visual search engines. One of the main problems with it comes to visual searching, is which feature we should extract ie. colour or material or relative size. These decisions could be seen as a typically "human" process, however, in visual search it is usually embarked upon by deep neural networks.

Deep Embeddings of neural networks allow the dimensionality of categorical variables to be lowered, as well as represent them meaningfully in an altered space. The uses of Deep embedding vary from similarity checks, retrieval of images and eliminating sparsity [4].

Visual features captured by cameras will vary for each image, so when clothes are photographed from different angles, the main goal is to minimize the differences in deep embeddings created for each image. Different angles make it harder to produce a consistent visual representation of an item, further making it difficult to compare. For example, Figure 1 shows a dress photographed from both front and back.

This paper explores techniques and models that allow deep embeddings to be learned from an image dataset, and hence drive visual search that can cope with multiple

views. With initial feature analysis in section III-B, followed by comparisons of popular image clustering models via deep embeddings in section III-A, we aim to provide an overall discussion on model accuracies, as well as identify tools and key features of each model. All code and data can found in our Github Repository [5].

II. Related Works

A large value of sales in e-commerce is driven by fashion and lifestyle sales with a user's buying decision primarily influenced by a product's visual appearance. With traditional text-based search engines focusing on textual metadata of products, the need for an accurate visual search engine is crucial to e-commerce sites, since detailed enough text metadata is difficult to find for each image and less standardized across images [6].

Many papers in the industry have evaluated the strength of different neural networks in the scope of visual searching. Pinterest, Alibaba, and SnapVIsion provide brilliant examples of neural networks trained so far, and our aim is to evaluate some of the fundamental techniques used in training their network.

Pinterest is a social media platform designed to help users find images that match their interests and preferences settings. By searching for images rather than text, users can upload an image or take a photo of an object, and Pinterest will show related points based on extracted features of the image. Their paper [7] begins introducing the importance of helping their users navigate through visual content - crucial to their social media service. They describe their multi-task deep learning system that learns a unified image embedding, which powers the visual search products used throughout their service. Emphasizing the importance of feature extraction from a wide variety of image sources, they have used an enormous amount of training data, whether it's high-quality camera images, scans, or product catalogs. Their products have then been used for both browsing and searching purposes in the realm of social media and visual search.

Alibaba is an e-commerce platform specializing in retail and technology, with a commitment to focus on the use of visual search to create better experiences for users [8]. Their primary visual search platform Taobao, allows users to upload images and find products that match. They use a large-scale visual search system infrastructure, that covers key challenges in the area, such as bridging the gap between user query images, and product catalogue images, and the difference in their features. Their aim to train Convolutional Neural Networks(CNNs) without large amounts of human annotation has shown the ways in which state-of-the-art deep learning techniques and deep embeddings are used to perform a visual search at large scale [9]. Furthermore, they understand the importance of improving user engagement via visual search, while still acknowledging the complexities of system development with regard to the types of images and data they are training with. The training of the CNN is something we touch upon in section IV-A2

Though the above examples have shown how these systems are made in scale, they still acknowledge the difficulty of the accuracy of visual searching. Further papers, such as Ionescu's paper [10] address the problem of estimating the difficulty of visual searching and image, in conjunction with human response times to identify images. They first gathered data through a crowd-sourcing platform and analyzed the impact of human interpretable image properties and their effect on visual search difficulties. Their model was able to correctly rank 75 % of pairs according to their difficulty score.

Although our paper discusses the importance of deep embeddings in visual search, it is important to note the wider applications of the technique. For example, visual searching and image clustering have been prominent in medicine [11].A deep learning-based visual search system for the task of automating the localization of masses in mammography images They split their system into two: a classification engine and a localization engine, with the former using deep embeddings. With a rate of 0.9 false positives per image, the true importance of the technique has been shown during the study.

III. Initial Data Exploration

As summarized in the introduction, this project is a Deep Embeddings-based multi-categorization task for images. Therefore, it might be a good idea to use the Convolutional Neural Network(CNN) to extract feature vectors of pictures in the original dataset, and then use these features as input to map the data to the embedding space.

A. Preparation

Based on the requirements of extracting feature vectors and image classification, a pre-train model Desnet201 is selected to train by our dataset. At the beginning of the training, "imagenet" is used [12], a pre-trained weight given by Desnet201 as default. Next, we tried to optimize the model with the data from our dataset to make it more compatible with our 1500 classification task. We train the model at a batch size of 128 for 200 Epochs through the Adam optimizer.

This fine-tuning avoids overfitting while better matching the characteristics of our image set. In the training process, we chose the Categorical Crossentropy as the loss function for the model because the labels are independent and mutually exclusive, and this is a multiple classification task. [13] In the meantime, we added the "ModelCheckpoint" function to avoid missing the best weights. [14]SoftMax is additionally added as the final activation function to get the classification results as output. This output will be used as a basis to get the Top5-acc, and then Top5-acc will be the criterion to detect the accuracy of the image feature extraction in Data exploration.

1) Data Quality: The data set provided was predominantly clean, with images ordered in a folder by item, and a set number of images per type of item. For example, a folder containing clothes has 5 images, while one with an accessory has 3 images. The images in the folder include multiple views of the dress, a photo of the model wearing the dress, and a close-up of a part of the dress. However, the content within each folder and the number of images differed, which we believe made it impossible to compare from folder to folder.

B. Data Exploration

At first, we randomly selected a few images in the dataset, the dataset was made up of 1500 different classes of clothing, each group consisting of a front view of the clothing and different angles photos of a model wearing the clothes.

For data exploration in image datasets, visualizing the extracted feature vectors by reducing their dimensions and clustering them while labeling typical points near the cluster center and then showing their corresponding original image is an effective visualization method.

Principle component analysis(PCA) is a commonly used statistical technique for reducing the dimensionality of high-dimensional datasets. At first, we planned to use PCA to reduce the dimension of the feature vectors from 1920 to 2, and then use k-means to cluster the 2D data. After that, we verified the classification results by taking random points many times. PCA has a good classification effect for relatively large items (such as sweatshirts, T-shirts, dresses, and shoes), but it is difficult to distinguish relatively small objects such as glasses and wallets. This is consistent with what we found at the beginning that the relatively high density of the points gathered in the red circle of the image. Small decorations were collectively reduced to this area. In order to solve the problem, we subdivide the dataset by increasing k. As a result, increasing the number of clusters cannot subdivide small items. Large items are more subdivided. For example, it separates the jacket from the sweatshirt class, which is affected by too low explained_variance_ratio (0.13940094 0.05814915) of the first two principal components after PCA.

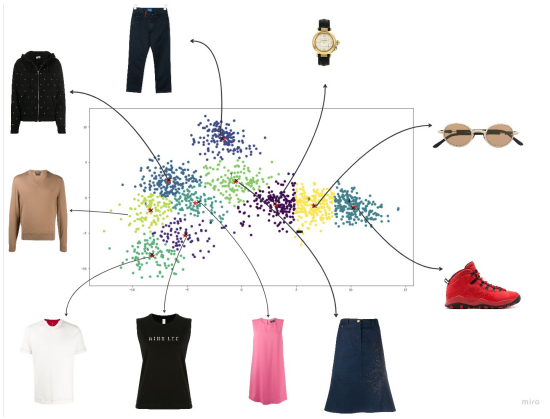


Fig. 2: PCA and Kmeans Classification results

Fig. 3: PCA and Kmeans Validation results

In order to solve the problem of subdivision of small items, a nonlinear dimensionality reduction method may be a good choice which is also reflected in the classification results. According to the results of picking multiple random verifications points many times, earrings, glasses, watches, and wallets are clearly distinguished with a gap with each other in this dimensionality reduction image as marked in the figure.

By using the feature given by Desnet201, we use the corresponding eigenvector projection to help identify the most important features or patterns in large datasets. Corresponding eigenvector projection is a technique used in linear algebra and data analysis to project data onto the eigenvectors of a covariance matrix. In this technique, the covariance matrix is first calculated from the dataset, and then its eigenvectors and eigenvalues are computed. The eigenvectors are then used to create a projection matrix that can be used to project the data onto a lower-dimensional space.

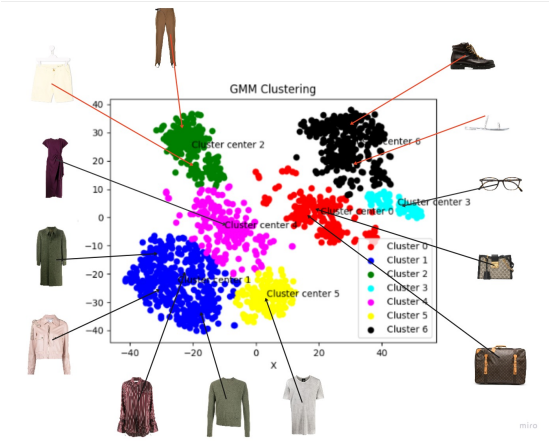


Fig. 4: t-SNE and GMM Classification results

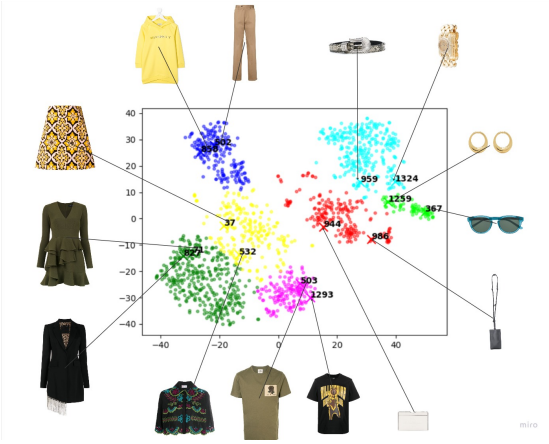
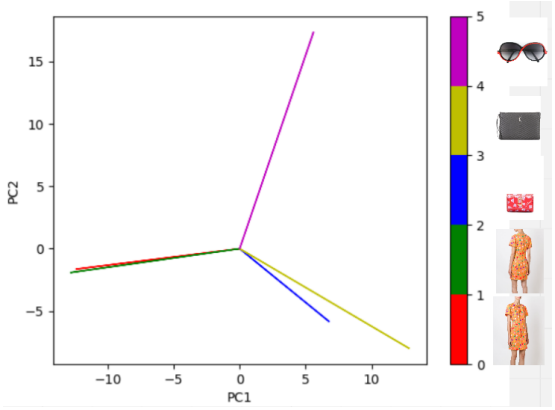
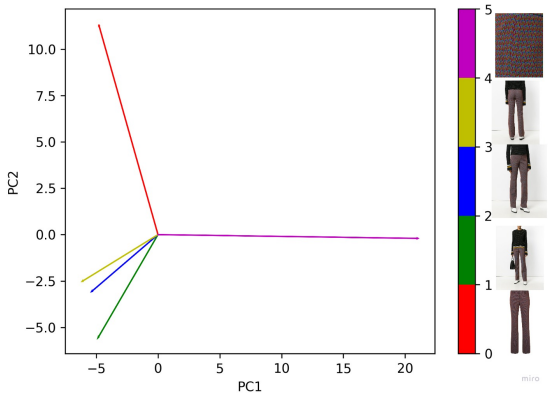


Fig. 5: t-SNE and GMM Validation results



(a) Distribution of corresponding eigenvectors for similar clothes.



(b) Incorrect vector projection in principal component analysis (PCA).

Fig. 6: Comparison of vector projections for clothing items.

The corresponding eigenvector projection technique is often used in principal component analysis (PCA). Principal component analysis was performed based on the obtained individual image features to obtain two main components. Corresponding eigenvectors are found in different images by using their features, which are projected onto the 2 axes. From the figure, it can be learned that in the images of the same clothes, such as this dress, the two vector distributions are relatively close to each other. In addition, the vectors of different items such as the bag and the dress distribute in different directions. And the 2 bags are distributed in a similar direction but with some differences.

However, when working with different view pictures of the same clothes. Sometimes, the situation as shown in the figure on the right occurs. The clothes from different viewpoints are projected in different two directions. We believe this is due to the fact that Desnet201 is too detailed in extracting features in separate images, resulting in a large gap in features between the results obtained when processing clothes from different viewpoints. These different features may come from the model's body shape, posture, and other clothes the model is wearing. That is the reason why the classification accuracy using Desnet201 dropped 45% for muti-view images, though with single-view images, the Top5-acc is 85%.

After analysis, it can be found that Desnet201, although it performs better in the feature analysis of different clothes, is less effective in the feature of different views of the same clothes. So we use the contrastive learning method to perform the analysis learning of clothing images.

IV. Models

A. Neural Networks in Deep Embeddings

1) Introduction: Our initial idea with modelling was to start off with a Convolutional Neural Network(CNN) [15] as the initial technique to classify images using our dataset.

2) Initial CNN Implementation : The overall method is as follows:

I. Feed the training data to the CNN
II. The model associates images and labels
III. The model makes predictions on the test set
IV. Verification of predictions from test labels

We split the data into training and testing sets and preprocess the data. Upon loading the files, images are resized, flattened to a one-dimensional array and accounting for pixel values of 0-255. The processed images and their labels are used during model training.

The building blocks of neural networks are layers, which extract representation from the fed into them. Each layer contains a number of nodes, where the node contains the current score indicating if the current image belongs to one of the classes.

We used the Keras [16] sequential model to specify the layers of the neural network. The first layer usually is to flatten the data, however, we had done this in preprocessing. The next two layers involved the dense layers. The dense layer is the fully connected neural layer. The first layer had 128 neurons/nodes and the second returned a logits array length of 6.

Before applying the model to train data, it was compiled using SparseCategoricalCrossentropy as loss function, optimizer (Adam) and metrics to monitor the training and testing steps.

Upon training on 50 epochs, the following accuracy and results were obtained:

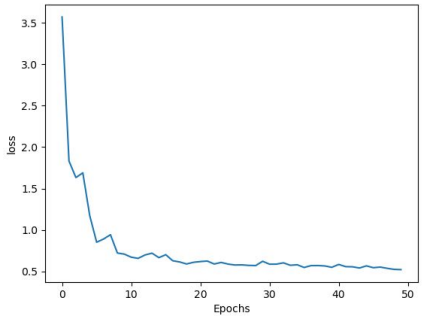


Fig. 7: loss CNN

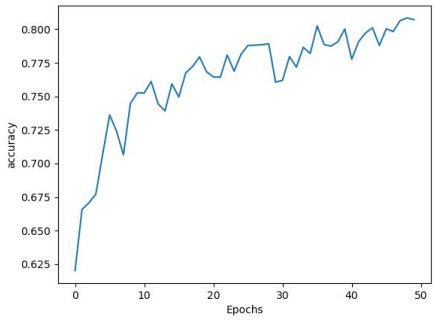


Fig. 8: accuracy CNN

The above figures show the accuracy and loss values vs the number of Epochs the model is run with. The accuracy for the test set predictions remains high at over 0.775 at around 50 epochs. With a loss value of around 0.5, CNN is arguably a good start to image classification, as shown via its frequent uses in the industry in visual search engines. The CNN is not computationally heavy and does not require big processing power, such as GPU usage.

In this figure, the bar represents how possible it is for the item to be classified within that label, derived from the model.

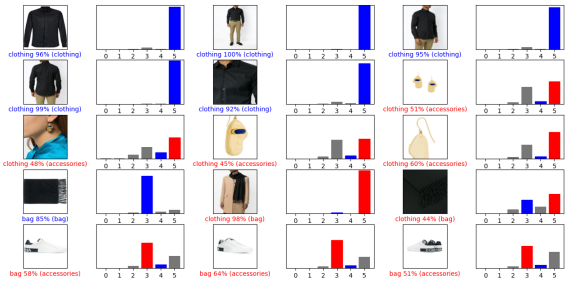


Fig. 9: Image classification

B. Transfer Learning & VGG Models

Transfer learning is the re-usage of weights of a pre-trained model, on a particular dataset/problem. The

premise behind transfer learning is that if a model is trained on a diverse and extensive dataset, it can be used as a general model for other visual tasks.

It has a multitude of applications in visual search via deep embeddings, with one of the most popular examples being VGG16 [17] and VGG19 [18], described below.

Being one of the most popular image recognition architectures, VGG acts as a baseline for many deep embedding models and has a standard deep convolutional network architecture as described below.

The first preprocessing layer takes an RGB image with pixel values in the range of 0-255 and subtracts mean image values. Upon preprocessing, the input images are passed through weighted layers. The training images are passed through convolution layers. In the VGG-16 model, there are 13 convolutional layers and 3 fully connected layers. In the VGG-19 model there are 16 convolutional layers with 3 fully connected layers. Both have the same 5 pooling layers integrated throughout as shown in Fig 10.

Fig. 10: VGG16 and VGG19 Layer Details. Image source: [19]

We ran both VGG16 and VGG19 with 50 epochs and two dense layers on top of the pre-trained VGG layer. We initially gave three classification labels: accessories, clothes, and homeware, however with little accuracy we increased to ten classes to subdivide them even further.

The plots in Figs 11 and 12 show accuracy and loss comparisons between the two VGGNet models.

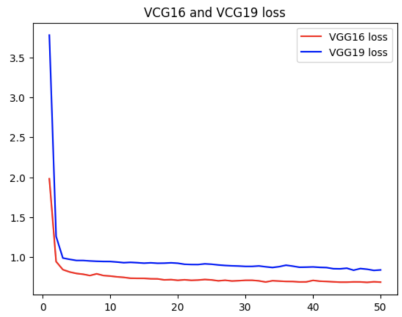


Fig. 11: VGG16 and VGG19 Loss

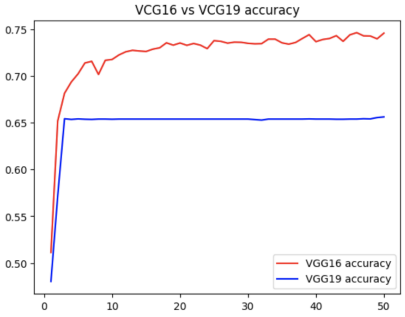


Fig. 12: VGG16 and VGG19 Accuracy

The loss for both models seems to be high, with accuracies lower than the loss in certain scenarios. Previous iterations of the models on other datasets [20] have shown around a 90% accuracy as it is usually regarded an advanced and accurate model.

A high loss value and a low accuracy value could be an indicator that the model is not fitting the training data correctly, with the following being potential reasons:

- **Insufficient data** - the model may not be able to learn underlying patterns in the data if the sample size is too small. This could potentially be the case with the sample we chose, with not enough of each class being represented
- **Overfitting** - the model may not generalise to the unseen testing if overfitting has occurred. This could be rectified with techniques such as regularization
- **Hyperparameters** - running the model with different hyperparameters would allow us to find optimal parameters such as learning rates or batch size
- **Preprocessing issues** - issues with preprocessing the data may have affected overall accuracy results, however, we believe this is less so the case as we used the requirements as specified in the VGG16 documentation [17].

C. Contrastive Learning

Contrastive learning is a technique used in deep learning that involves the creation of a contrastive loss function. The goal is to learn a feature representation for a set of data in a way that similar samples are mapped to nearby points in the feature space, while dissimilar samples are mapped to distant points. This approach has been used in several applications, including image classification and retrieval tasks [21].

In this report, we utilized a contrastive learning loss function to learn deep embeddings for clothing items to enable visual search that can detect different angles and viewpoints of the same items of clothing. Specifically, we used the loss function proposed in [22] and [23], which have shown promising results in learning discriminative features for visual search tasks.

To employ contrastive learning with the ResNet-34 model, we used transfer learning, a technique that allows us to reuse pre-trained models and fine-tune them for our specific task. The pre-trained ResNet-34 model was used as a backbone network, and the contrastive learning loss was added on top of it to learn meaningful representations of clothing patterns and styles.

Due to the absence of data containing specific clothing styles, we utilized a self-supervised contrastive learning approach. By augmenting the images, and pairing two augmented versions of each image while simultaneously pushing apart the representations of different images, we aimed to group similar categories. Examples of these augmentations can be seen in Fig 13, where the images have been randomly transformed to create slightly modified versions of the original ones.

During training, the ResNet-34 model was fed pairs of images, one from the same clothing category and another from a different category. As a result, the model learned to minimize the contrastive loss function, which is a mathematical representation of the distance between embeddings of different categories and the similarity between embeddings of the same category. Through this process, the model learned to distinguish between various clothing patterns and styles.

The results showed that our model had a promising ability to group together clothing patterns and styles, see Fig 14. However, the groups were also reliant on color, as we chose not to include any color adjusting augmentations during training. We made this choice because we assumed that a visual search clothing style model should group together clothing colors. Future work could explore augmentations that allow the model to learn more about the relationship between color and clothing style. Additionally, it's worth noting that computing accuracy for

Two augmented versions of a shoe.

Two augmented versions of a smart outfit.

Fig. 13: Two example clothing items and their corresponding pair of augmented samples.

self-supervised contrastive learning models is not feasible due to the absence of "true" categories. These models learn by contrasting the similarities and differences between different samples rather than assigning them to pre-defined categories. Therefore, while we can evaluate the model's ability to group similar samples together, we cannot measure its accuracy in the traditional sense.

In conclusion, the use of transfer learning and contrastive learning allowed us to create a powerful model for visual search clothing style. The model could effectively group clothing items with similar patterns and styles, and with additional augmentations, could become even more robust. This research has the potential to impact the fashion industry by providing a new tool for customers to visually search for clothing items that match their style.

V. Findings

Fig. 14: Results for the Self-supervised Contrastive Learning Loss. The lower the similarity score, the closer the image is to the query.

A. Overall Conclusions

The table below shows a comparison of the accuracy scores between the models run. Possible reasons for a low score have been stated above, the table still provides an insight into different accuracies:

	CNN	VGG16	VGG19
Loss	0.62	0.59	0.6
Accuracy	0.7	0.72	0.79

In conclusion, we explored a variety of models and methods for feature extraction, as well as explored models for image classification. Although not all models were able be compared by their accuracy, for example the Self-supervised Contrastive Learning Model, the methods used and tested, as well as the reasoning behind them proved to be useful in the overall context of the problem.

Furthermore, we were able to investigate further applications of models such as the use of VGG in the DEC model, overall providing a user with an investigation into the techniques that drive visual search engines they use in day-to-day life as well as provide useful future work such as an implementation of the DEC model via VGGNet.

B. Future Works

Transfer learning is a method used in deep learning where a neural network model is initially trained on a large classification problem, and then applied to a different problem. The basic idea behind transfer learning is that if a model is trained on a diverse and extensive dataset, it can be used as a general model for other visual tasks.

Earlier, the VGG16 and VGG19 models were run and compared. A popular extension of this transfer learning model is the Unsupervised Deep Embedding for Clustering(DEC) Model.

The DEC model simultaneously learns feature representations and cluster assignments using deep neural networks. It learns a mapping from the data space to a lower-dimensional feature space in which it iteratively optimizes the cluster assignment and the underlying feature representation.

DEC has two phases:

- Phase one: parameter initialization with a deep auto-encoder
- Phase two: parameter optimization (in our case clustering) where it iterates computing a target distribution and minimizes a divergence measure

In the future, we would extend the VGG16 and VGG19 models we made to apply to the DEC model, running it on our dataset to compare to the basic transfer learning models implemented, hence evaluating further effective methods for image clustering via deep embeddings.

C. Ethical Considerations

For our dataset, we found to have few serious ethical problems, however, the stealing of personal data/impersonating of data or the uploading of accidentally incorrect images could be an ethical issue. Main ethical issues would arise if datasets were being shared with a lack of consent.

However, when the models described are applied to a wider range of topics, further ethical issues could arise. For example, in the medical paper [11] mentioned previously, ethical issues such as an issue with privacy could relate to the storage and analysis of individuals' photos could be considered. Patients should always give consent to the storage of these photos for the purpose of running visual searches to check for masses. Another problem is bias in the use of the images, as it should represent the broader population and not just a subset of patients.

D. Reflective Discussion (Quincy Sproul - JL20645)

My primary responsibility as a member of the project team was to research and implement the contrastive learning model for visual search clothing style. My objective was to enable the model to detect different angles and viewpoints of the same clothing items, grouping similar categories and distinguishing between various clothing patterns and styles.

During the project, I encountered a significant challenge due to the absence of data containing specific clothing styles, this meant that in order to group together "clothing styles" I had to come up with my own interpretation of said styles. To address this obstacle, I utilized a self-supervised contrastive learning approach. I augmented the images and paired two versions of each image while pushing apart the representations of different images. Through this process, I aimed to group together similar categories.

Another issue that I encountered was the model's dependence on color. I made a conscious decision not to incorporate any color adjustment augmentations into the training process, believing that a visual search clothing style model should naturally categorize clothing colors. Nonetheless, future research may consider employing new techniques to help the model better understand the interplay between color and clothing style by using transformations such as random color jitter or random contrast.

In spite of these obstacles, the outcomes of our study demonstrate the potential of the model to accurately group clothing patterns and styles. By leveraging transfer learning and contrastive learning techniques, I were able to develop a powerful visual search clothing style model that may become even more resilient with further augmentations.

During the project, I faced a challenge with the quality of the image dataset we were using. Some of the image samples from identical classes were very different in terms of viewpoints, dominant colours and lighting conditions. This variability in the dataset caused difficulties in training the model to accurately detect and group similar clothing patterns and styles. In order to overcome this challenge, I employed data augmentation techniques to generate additional training data that covered a wider range of viewpoints and lighting conditions. I also used techniques such as normalization to improve image quality and reduce variability across similar classes of clothing. These steps helped us create a more robust and accurate model that was better able to handle the variability in the dataset.

In conclusion, my contribution to the project was essential in creating a model that could effectively group clothing items with similar patterns and styles. By applying my knowledge of deep learning and contrastive learning techniques, we achieved our project's goals. Throughout the project, I gained valuable experience in implementing machine learning models, working with a team, and overcoming challenges. I am proud of our team's achievement, and I believe that our research has the potential to make a positive impact on the fashion industry.

References

[1] Alasdair DF Clarke, Micha Elsner, and Hannah Rohde. Where's wally: The influence of visual salience on referring expression generation. *Frontiers in psychology*, 4:329, 2013.

[2] Clark Boyd. The Past, Present, and Future of Visual Search. https://medium.com/swlh/the-past-present-and-future-of-visual-search-9178f006a985, 2018.

[3] Clark Boyd. Visual Search Trends: Statistics, Tips, and Uses in Everyday Life. https://clarkboyd.medium.com/visual-search-trends-statistics-tips-and-uses-in-everyday-life-d20084dc4b0a, 2018.

[4] How Machine Learning Embedding Works: A Complete Guide. https://analyticsindiamag.com/machine-learning-embedding/, Unknown.

[5] Lipi Chakraborty, Siana Dicheva, Quincy Sproul, Chenghao Zhang, Yufeng Yang, and Weiguang Wang. Ads. https://github.com/SianaDicheva/ADS, 2023.

[6] Devashish Shankar, Sujay Narumanchi, H. A. Ananya, Pramod Kompalli, and Krishnendu Chaudhury. Deep Learning based Large Scale Visual Recommendation and Search for E-Commerce. http://arxiv.org/abs/1703.02344, 2017.

[7] Andrew Zhai, Hao-Yu Wu, Eric Tzeng, Dong Huk Park, and Charles Rosenberg. Learning a Unified Embedding for Visual Search at Pinterest. http://arxiv.org/abs/1908.01707, 2019.

[8] Yanhao Zhang, Pan Pan, Yun Zheng, Kang Zhao, Yingya Zhang, Xiaofeng Ren, and Rong Jin. Visual Search at Alibaba. https://arxiv.org/abs/2102.04674, 2021.

[9] Sen Li, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, Xiao-Ming Wu, and Qianli Ma. Embedding-based Product Retrieval in Taobao Search. https://arxiv.org/abs/2106.09297, 2021.

[10] How Hard Can It Be? Estimating the Difficulty of Visual Search in an Image. https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Ionescu_How_Hard_Can_CVPR_2016_paper.html.

[11] Mehmet Günhan Ertosun and Daniel L. Rubin. Probabilistic visual search for masses within mammography images using deep learning. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1310–1315, 2015.

[12] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[13] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[14] Jostein Barry-Straume, Adam Tschannen, Daniel W Engels, and Edward Fine. An evaluation of training size impact on validation accuracy for optimized convolutional neural networks. *SMU Data Science Review*, 1(4):12, 2018.

[15] Basic classification: Classify images of clothing. https://www.tensorflow.org/tutorials/keras/classification.

[16] Keras documentation of Convolution layers. https://keras.io/api/layers/convolution_layers/".

[17] VGG16 documentation. https://pytorch.org/vision/main/models/generated/torchvision.models.vgg16.html.

[18] VGG documentation. https://keras.io/api/applications/vgg/.

[19] Fei-Fei Li, Justin Johnson, and Serena Yeung. Lecture 9: CNN Architectures. http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture09.pdf, 2019. Accessed on May 4, 2023.

[20] VGG19 on fashion-mnist. https://github.com/khanhnamle1994/fashion-mnist/blob/master/VGG19-GPU.ipynb.

[21] Alaaeldin El-Nouby, Natalia Neverova, Ivan Laptev, and Hervé Jégou. Training Vision Transformers for Image Retrieval. https://arxiv.org/abs/2102.05644, 2021.

[22] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning. https://arxiv.org/abs/2004.11362, 2020.

[23] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. https://arxiv.org/abs/2002.05709, 2020.