DL4DS Contrastive Learning on IMDb Faces Inspired by SimCLR

Daniel Foley

January 26, 2024

Abstract

Performing Self-Supervised Representational Learning (specifically Contrastive Learning) on the IMDB faces dataset to construct a model that determines appearance similarity between human subjects.

Introduction

My interest in embeddings generation paired with my interest in self-supervised learning led me in the direction of contrastive learning when beginning to think about my project. I decided to use the dataset IMDb-Faces, which contains 460,000 cropped images of celebrities scraped from IMDb, including metadata on the person within each image. With this dataset of faces, I was influenced by SimCLR, a contrastive learning approach, to create an embeddings model for images that determines facial similarity.

Related Work

Ercisson et al. released a paper which covered broadly the task of Self-Supervised Representation Learning, describing techniques that were relevant as of 2021 [1].

Chen et al. released a paper which discussed their new BagSSL approach to SSL image representation learning which suggested improved results when tested on ImageNET [2].

Uelwer et al. released a paper which also covers Self-Supervised Representation Learning techniques and recommendations [3].

Chen et al. released a paper proposing a Contrastive Learning appraoch called SimCLR, where they were able to perform Self-Supervised Learning on ImageNet and achieve notable results[4].

Wang et al. released a paper that constructed a noise-controlled dataset of human faces using IMDB and performed analyses. [5]

Approach

SimCLR

In the paper "A Simple Framework for Contrastive Learning of Visual Representations", a proposed framework for contrastive learning is trained on ImageNet, generating successful results.
I found SimCLR to be an interesting first step into contrastive learning, so it is what I based my project around.

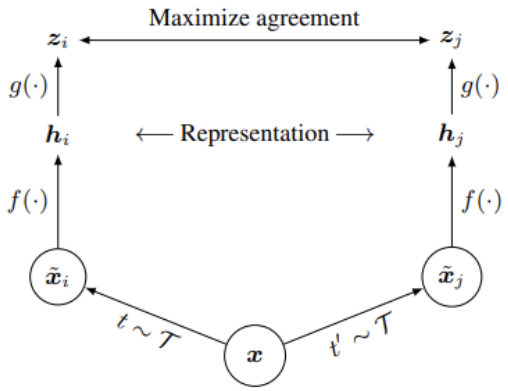

Figure 1: Overview: For each image (x) in the dataset: create two new images(x_i,x_j) with a set of differing augmentations (t,t'), then encode ($f(\cdot)$) all images, perform feature extraction (h_i,h_j), reduce the dimensionality of embeddings ($g(\cdot)$), and apply a contrastive loss function to train the encoder.

Data Processing

I used pandas to filter image metadata for candidate selection for my training set. I decided to keep my training set at least 100k images. My final filters that I applied in an attempt to maximize image inclusion and performance were the following:
1. Only include images of individuals with at least 50 images in the dataset to prevent class imbalance when testing.
2. Only include images with at least 100x100 resolution, as I qualitatively found that any lower resolutions resulted in a lack of distinct features, even when upscaled and interpolated.

Notably, something I would do differently here would be to exclude images where no face was detected in the facial similarity metric from the metadata and make up for that reduction in images by including all individuals within the dataset. This would have reduced outlier images whilst still building a representational understanding of human faces.

Once selecting candidate images, I sampled each class (name) for images to select 50 images for each name to enter the training set, again in the misguided interest of class balance. This resulted in about 100k images.

After developing a training set, I performed scaling on them to normalize all the images to 224x224 resolution, which is what SimCLR used. While upscaling images, I applied an interpolation of pixel values to mimic an increased resolution.

I also created a testing set by sampling images associated with each name in the training set excluding the selected training images. Each class was sampled 10 times, resulting in a cumulative body of about 20k images.

Model

For the transformation step, instead of sampling two augmentations like in SimCLR, I randomly applied a number of transformations including random cropping, horizontal flips, color jitter, grayscale and gaussian blur. The goal of this was to modify the image more than in SimCLR for better generalizability, though this could have also resulted in certain features being transformed out.

For the encoder, I decided on a pre-trained ResNet-18, which is smaller than the ResNet-50 from SimCLR, the intent was to reduce training time given the schedule of the class. The ResNet includes the pooling of step h and so is immediately fed into the projection head.

I then used the NT-Xent loss function used in SimCLR to calculate contrastive loss and train the encoder.

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)}$$

Training

For training, I trained 100k base images, thus 200k total images, for 50 epochs using a batch size of 128. The specific use of 50 epochs and 128 batch size were both in the interest of time relative to this project's deadline.

Whilst training, the model saves checkpoint versions every 10 epochs, with the intent to be that I could perform validation on each one in retrospect to visualize performance and potentially start from that version if training fails for some reason.

Datasets

I used the cropped IMDb-Faces dataset available at
Link to Paper

I also planned on using the metadata to evaluate performance of the model, however that did not turn out as I will cover in Evaluation.

The IMDb-Faces dataset contains 460,723 scraped images from IMDb, with the cropped dataset reducing the frame of each image to only be the face of the person associated with it in the metadata.
The Metadata includes the following fields:
dob: date of birth (Matlab serial date number)
photo_taken: year when the photo was taken
full_path: path to file
gender: 0 for female and 1 for male, NaN if unknown
name: name of the celebrity
face_location: location of the face.
face_score: detector score (the higher the better). Inf implies that no face was found in the image and the face_location then just returns the entire image
second_face_score: detector score of the face with the second highest score. This is useful to ignore images with more than one face. second_face_score is NaN if no second face was detected.
celeb_names (IMDB only): list of all celebrity names
celeb_id (IMDB only): index of celebrity name

Evaluation Results

My evaluation results were largely focused around generating embeddings for all test images, then computing cosine similarity to determine like and unlike images for retrieval of k images.

My initial plan was to evaluate the model by retrieving 9 embedded images for each embedded image in my test set and averaging the percent that were from the same individual as the target image. With the ultimate goal of being able to perform KNN on an image outside the dataset to determine the most similar celebrity from within the dataset.

Upon qualitative analysis of the performance of my model on retrieval, I observed that target images almost never retrieved the same person, instead, it was often images similar to the target. This invalidates my evaluation approach and limited me to strictly qualita-

tive analysis.

Figure 2: Example retrieval in which the target image was Al Pacino (top), however neither retrieved image was from the same actor

Notably, neither of the retrieved images are Al Pacino, however, qualitatively note the similarity in jawline, nose, and eyes. While technically the model did not perform as intended for evaluation, there is evidence of image similarity.

Another observation I noted while combing through retrievals was the tendency of the model to retrieve images with similar silhouettes. This means that retrieved images of faces had a tendency to all be looking a similar degree away from the camera. As evidenced in Figure 2, note how both retrieved images portray men slightly tilting their head down and looking just off-center of the camera whilst still facing the camera. Obviously, this is unintended behavior, however it is indicative of similarity being computed properly.

Conclusion

In summary, this paper catalogs a contrastive learning approach influenced the architecture of SimCLR on the IMDb-Faces dataset, with some alterations made in model architecture in the interest of chronological scope of the project as well as the different nature of the IMDb-Faces dataset from ImageNet (What SimCLR was originally trained and evaluated on).

Continued work could be done in varying the size and architecture of the encoder, refinement of training data selection and pre-processing, and increasing dataset scale by performing broader scraping. I would especially be interested in observing how the traditional SimCLR technique and architecture measures against the performance of this paper's more lightweight approach.

Ultimately, while not performing directly as intended, the performance of the trained model presented indications of success towards the interest of contrastive representational learning resulting in a satisfactory conclusion.

References

[1] Ericsson et al. *Self-Supervised Representation Learning: Introduction, Advances and Challenges*. arxiv, 2021.

[2] Chen et al. *Bag of Image Patch Embedding Behind the Success of Self-Supervised Learning*. arxiv, 2022.

[3] Uelwer et al. *A Survey on Self-Supervised Representation Learning*. arxiv, 2023.

[4] Chen et al. *A Simple Framework for Contrastive Learning of Visual Representations*. arxiv, 2020.

[5] Wang et al. *The Devil of Face Recognition is in the Noise* arxiv, 2018.