Identical Image Retrieval using Deep Learning

SAYAN NATH[1], NIKHIL NAYAK[2]

[1, 2] *School of Computer Engineering, Kalinga Institute of Industrial Technology, India*

Abstract— In recent years, we know that the interaction with images has increased. Image similarity involves fetching similar-looking images abiding by a given reference image. The target is to find out whether the image searched as a query can result in similar pictures. We are using the BigTransfer Model, which is a state-of-art model itself. BigTransfer (BiT) is essentially a ResNet but pre-trained on a larger dataset like ImageNet and ImageNet-21k with additional modifications. Using the fine-tuned pre-trained Convolution Neural Network Model, we extract the key features and train on the K-Nearest Neighbor model to obtain the nearest neighbor. The application of our model is to find similar images, which are hard to achieve through text queries within a low inference time. We analyse the benchmark of our model based on this application.

Indexed Terms-- Image Similarity Search, Computer Vision, Deep Learning, Transfer Learning, BigTransfer

I. INTRODUCTION

How can we compute the similarity of one image to another? This is a question that has been asked for hundreds of years, and it is probably the most fundamental question in image processing. Several customer-facing applications leverage images to search and find products and they usually complement that of a text-based search in most use cases. In this paper, we will describe an approach to compute image similarity using deep neural networks. Our method is based on a *state-of-the-art* model known as the *BigTransfer* Model, which learns to predict the similarity of two images. Fig 1. Shows examples of Image samples from the *tf_flower* [4] dataset on which our *BigTransfer(BiT)* [1] model is trained.

Our system has the potential to be used with business-critical applications. With the visual exploration trend rising in the retail sector and the availability of quintillion Gigabytes of data at our disposal our image similarity model is bound to become more and more accurate. We optimized our model for two specific objectives: search accuracy and query duration. We achieved high accuracy by examining a few other candidates. We achieved a query time of under a second. Our system is highly accurate and fast. Our system is fast because we used a few other models such as hand-crafted features, autoencoders[11], CNN's[8] and we only used the most recent pre-trained model.

This approach is highly robust and we believe it could be adopted by the industry. We chose a specific type of deep learning model because of the best performance we have observed and because of the clarity of the approach. The state-of-the-art model we chose, based on the results, can leverage unsupervised learning. With that type of model, it is easier to train data in massive quantities and to have a much better generalization. BigTransfer(BiT) [1] model uses a feed-forward layer over an unsupervised autoencoder[11] along with an attention model \cite{Attention}. Our paper reports the first use of an attention model \cite{Attention} for model-based image retrieval. Model-based image retrieval has been used for a long time in text-based searches for making more accurate queries. But image-based search and image retrieval are very different scenarios. Visual perception works differently than text-based search. So is image-based search and image retrieval. Many other applications besides those highlighted in this paper can be benefitted from this work. Our system is robust, highly accurate, and it can leverage unsupervised learning to produce large amounts of high-quality training data. This has enormous potential in several domains.

While image search engines like Google or Bing can find similar images with a text search, they are not very effective for a variety of reasons. The most obvious reason is that textual descriptions are limited to 20,000 characters and cannot express detailed semantics or the history of an object. Secondly, people who don't speak English cannot query image search engines via language translation. This brings about the second problem of a large and growing number of users from emerging markets who may not have access to Google and Bing. More importantly, they do not have a search query language of their own, yet still, they express their need to find images of stars and moons using such complex terms that require a series of complex symbols or gestures to communicate. Indeed, the search relevance of image search engines is heavily dependent on the existence of similar images available on the Internet. This is an issue for all images that are not publically accessible for use, images captured at different times, or cameras with varying parameters. This is a very important point that we take into consideration when working on this project. On one side we have images that are not publicly accessible. The quality of the images will be varied and they will require careful content selection in the selection and processing of training and test data. On the other side, there is a lot of commercial information that "needs to be made publicly available for commercial use but is not yet. It can be difficult for a company to get permission to use their data, and another difficult process to get the images in the public domain. We also expect to expand and improve our existing models to make the selected publicly available images accessible for future work. However, in this project, we are primarily focusing on developing our data collection pipeline to collect images that are currently out of the public domain and publish them on our existing system. The project's goal is to assist small and medium enterprises to more easily produce a high-quality dataset in the image and document domains. The paper has used all possible efforts to make our project as replicable and reproducible as possible.

So far we have classified on flower dataset [4] and we have employed an unsupervised feature extraction technique for image preprocessing. In the future, we will release the new datasets, which can be useful to anyone else for training and deploying their system for image retrieval. Also in the future, we will explore a method to integrate our images of objects into deep search engines to improve the depth of the search results. This will add some search value, which is useful to some specific applications such as content discovery on an exhibition. We expect that this will drive towards some additional use cases in the future. Research and industry have benefited enormously from advances in digital communication technology. This technology has enabled tremendous growth in markets such as mobile commerce, social networking, and large-scale community projects. In this project, we envisage that such image-based platforms, which focus on image-based search \cite{SIR, deepcnn}, discovery, and mobile e-commerce, will present a massive opportunity for new business models.

II. RELATED WORK

A. Similar Image Retrieval (SIR)

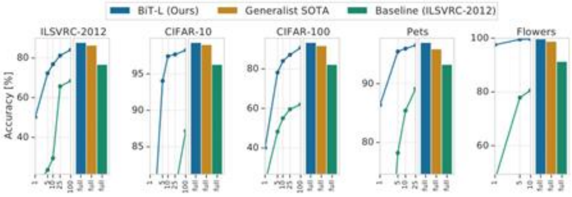
The central idea is to convert each image into a fingerprint, signature, or unique descriptor [5]. Internally, fingerprints are basically embeddings calculated from a suitable deep neural network. We have tried many generations of embedding techniques and adopted VGG16[12] as our primary network. The embedding is removed from the last fully connected layer of VGG16[12]. A typical embedding is a high-dimensional vector consisting of floating-point numbers. In the binarized format, the embedding is broken down into smaller subcodes that are included in the Elasticsearch index. When searching, we also use Elasticsearch's ability to perform efficient Hamming distance calculations in the form of bit-wise operations. Business applications are primarily focused on images that have been added to the catalogue in the last few weeks. Therefore, design the indexing process as a rolling process to constantly index new and recently updated images. The currently deployed system listens for Kafka themes that stream new and updated images. The running index (last 3 months) of the newly created image is maintained for later search and retrieval [5]. Due to the continuous nature of the application, hash-based indexing is the preferred choice over the technique of collectively learning representations from static datasets such as Principal component analysis (PCA) [3]. As the catalogue changes, so do the best key components, which require frequent recalculations. During the search phase, the query image (also known as the seed image) is provided to the system via the front end. On the backend, the query image is converted to an embed and its closest neighbour [2] is retrieved from the indexed store. The retrieved images are displayed in the grid in descending order of similarity to the query. Each resulting image has a checkbox that allows the user to select only the relevant image from the grid.

B. Image similarity with Deep CNN

Image validation algorithms aim to determine if a particular pair of images is similar [7]. Image verification is not similar to image identification. The former solves similar image use cases, while the latter is due to the nature of image retrieval. Advances in image verification relate to two main areas: image embedding and metric learning [27]. Image embedding trains robust and discriminating descriptors to represent each image as a compact feature vector/embedding. The current state-of-the-art function descriptor is generated by the self-learning function CNN[8]. SimNet [23] uses multiscale CNN[9] in a Siamese network [22] that learns 4096-dimensional embedding of images. Pairing is required for the Siamese network [22]. A pair of positive images (almost similar images) and a pair of negative images (different images). This allows you to learn the range of intervals. It turns out that choosing the right image pair for training is very important for achieving good model performance and faster model convergence. We propose a new online pair mining (OPMS) strategy that attempts to steadily increase the difficulty of image pairs during network training. Multiscale CNN[9] is used in Siamese networks. This CNN[8] learns common image embeddings in the upper and lower layers. This model learns much better image embedding than traditional CNNs[8] for image similarity tasks.

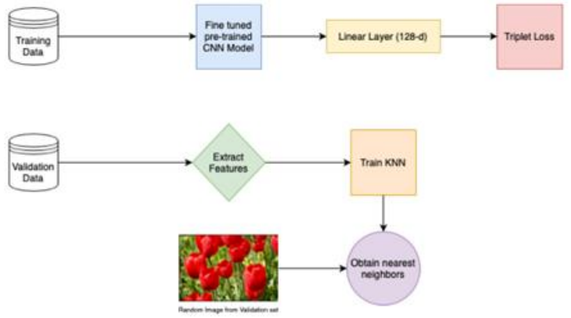
III. PROPOSED METHOD

We have used BigTransfer(BiT) [1] as our pre-trained CNN model[8] to extract the features. BigTransfer(BiT) [1] is a set of pre-trained image models that can be transferred to obtain excellent performance on newer datasets, even with few examples per class. The performance of the BigTransfer(BiT) [1] model increases as the dataset size increases. Transferring of pre-trained representations improves the efficiency and simplifies hyperparameter tuning when we train a deep neural network for computer vision. BigTransfer(BiT) [1] is trained in a large, generic dataset and its weights are then used to initialise subsequent tasks that can be solved with fewer data points, and less computation. BigTransfer(BiT) [1] is trained on three large datasets i.e JFT-300M dataset [6], which contains 300M noise labelled images. BigTransfer(BiT) [1] is transferred to execute many diverse tasks by setting the train set sizes from one example per class to one million examples per class. BigTransfer(BiT) [1] is highly effective and provides insight into the interplay between scale, architecture and training hyperparameters.



The x-axis shows the number of images used per class, ranging from 1 to the full dataset. On the plots on the left, the curve in blue above is our BiT-L model, whereas the curve below is a ResNet-50 pre-trained on ImageNet (ILSVRC-2012)

We are extracting the BigTransfer(BiT) [1] model to extract the image features and using those features we are calculating the distance between the images using the reference image. The greater the distance the more likely the images are alike. It scales well to millions of data points, and It achieves a good trade-off between the probability of error and the number of data points needed to achieve this probability.



IV. EXPERIMENTS

A. Experimental Setup

In this experiment, we have used the dataset of flowers [4] provided by tensorflow. It is popularly known as \texttt{tf_flower} [4]. TensorFlow Flower Dataset [4] consists of five classes. Five classes are labelled as 'Daisy', 'Dandelion', 'Roses', 'Sunflowers', 'Tulips'. The number of classes for the dataset is imbalanced.

The dataset is being loaded using the tensorflow dataset [4]. The dataset is then divided into train and validation data. 85\% of the data is taken as the training data and the rest of the data is taken as validation data. The number of training and validation samples were 3120 and 550. To make our model more robust I applied data augmentation \cite{augment, augmentation}. Used random flip with horizontal and vertically on the images. Used a random rotation with a factor of 0.2. Applied random zoom at a height and width factor of 0.2. Training images are resized to 160 and images are randomly cropped to 128. Validation images are resized to 160.

$$||f(x^a{}_i) - f(x^p{}_i)||^2{}_2 < ||f(x^a{}_i) - f(x^n{}_i)||^2{}_2$$

$$\text{for i in range(m)}$$

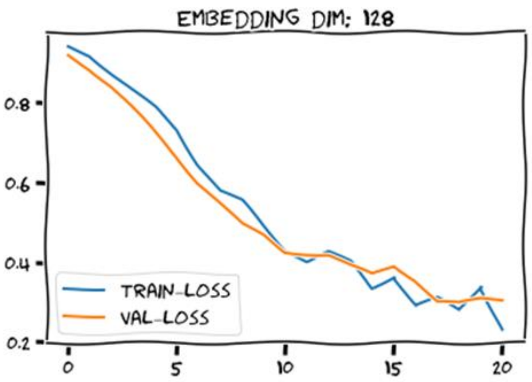
$$\Theta j = \Theta j - \alpha(\hat{y}^i - y^i)x_j{}^i$$

We loaded the pre-trained BigTransfer(BiT) [1] model which is trained on ImageNet21k[13] downloaded from TensorFlow Hub [14][15][16]. We created a BigTransfer(BiT) [1] model and normalised the dense representation. Used TripletSemiHardLoss [17] as a loss function. The loss encourages the positive distances (between a pair of embeddings with the same labels) to be smaller than the minimum negative distance among which are at least greater than the positive distance plus the margin constant (called semi-hard negative) in the mini-batch. If no such negative exists, use the largest negative distance instead.

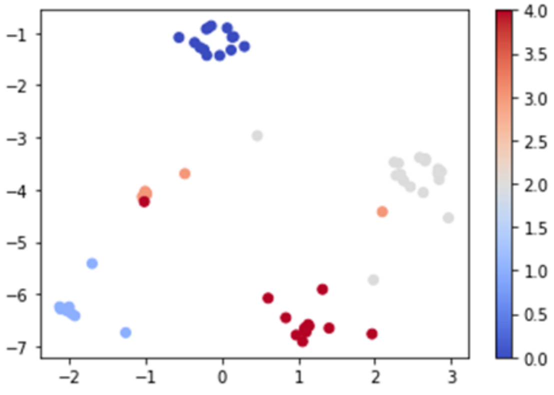
After compiling the model and the callbacks for the models are set up. Used Early Stopping to monitor the validation loss with a patience rate of 5 and standard rate for verbose i.e 2. Used CSV Logger to log all the data during training the model.

B. Experimental Results

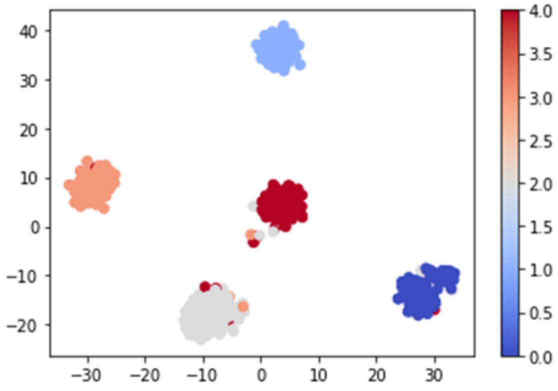
After setting up the experiment, we trained our BigTransfer(BiT) [1] model on the flower dataset [4]. Our training started with a loss of 0.94 and a validation loss of 0.91. The training is being done on Tesla V100-SXM2. The average time for an epoch is 2 seconds. The training time was 78.68 seconds with only 21 epochs. Our loss and validation loss after 21 epochs is 0.23 and 0.30 respectively.



a) Training Graph

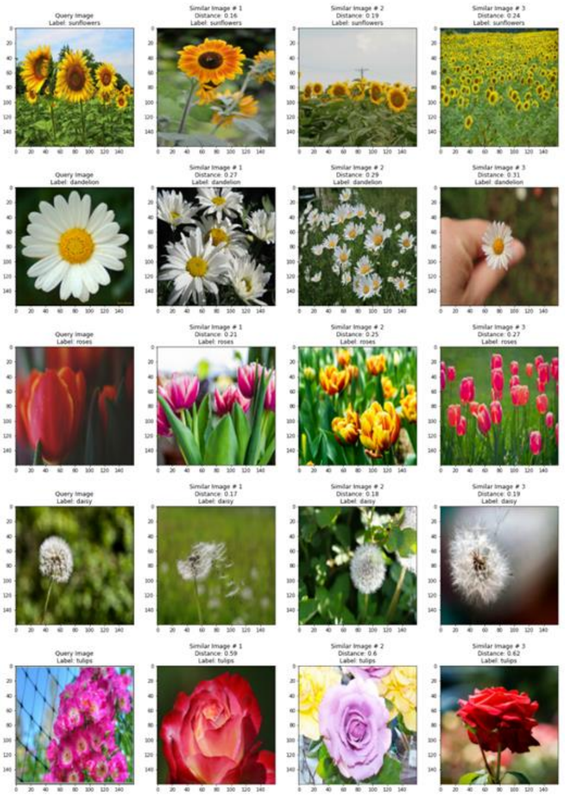


b) Visualizing the Embedding Space for Current Validation Batch



c) Visualizing the Embedding Space for Entire Validation Batch

Saved the BigTransfer(BiT) [1] model after training the model on flower dataset [4]. Defining the BigTransfer(BiT) Model to load the model weights we saved as our Keras model. Created a validation pipeline for training the nearest neighbor [2] model. We are calculating our nearest neighbor [2] for the features of our query image. The model took 0.00043 seconds for 550 samples.



CONCLUSION

This project shows how to scale up the pre-trained model by training on a larger dataset to extract key features. The model is based on a Resnet152[10] backbone and the classifier is an FCN-8s. The model is based on the code release for BiT-M. BigTransfer(BiT) [1] does not require any pre-processing to the input images, i.e., we can use vanilla ImageNet models to transfer on any visual tasks. We see this as an important step towards practical implementations. The idea behind our model is to build highly accurate and low latency visual search tools that can fit in with many business facing applications.

REFERENCES

[1] Kolesnikov A. et al. (2020) Big Transfer (BiT): General Visual Representation Learning. In: Vedaldi A., Bischof H., Brox T., Frahm JM. (eds) Computer Vision – ECCV 2020. ECCV 2020.

Lecture Notes in Computer Science, vol 12350. Springer, Cham. https://doi.org/10.1007/978-3-030-58558-7_29.

[2] Padraig Cunningham, Sarah Jane Delany. k-Nearest Neighbour Classifiers: 2nd Edition (with Python examples). arXiv preprint arXiv:2004.04523, 2020.

[3] Felipe L. Gewers, Gustavo R. Ferreira, Henrique F. de Arruda, Filipi N. Silva, Cesar H. Comin, Diego R. Amancio, Luciano da F. Costa. Principal Component Analysis: A Natural Approach to Data Exploration. arXiv preprint arXiv:1804.02502, 2018.

[4] TensorFlow Dataset. https://www.tensorflow.org/datasets/catalog/tf_flowers.

[5] Theban Stanley, Nihar Vanjara, Yanxin Pan, Ekaterina Pirogova, Swagata Chakraborty, Abon Chaudhuri. SIR: Similar Image Retrieval for Product Search in E-Commerce. Accepted in 13th International Conference on Similarity Search and Applications, SISAP 2020 arXiv preprint arXiv:2009.13836, 2020.

[6] Sun,C.,Shrivastava,A.,Singh,S.,Gupta. A.:Revisiting unreasonable effectiveness of data in deep learning era. In ICCV (2017) arXiv preprint arXiv:2009.13836, 2017.

[7] Srikar Appalaraju, Vineet Chaoji. Image similarity using Deep CNN and Curriculum Learning. arXiv preprint arXiv:1709.08761, 2017.

[8] Keiron O'Shea, Ryan Nash. An Introduction to Convolutional Neural Networks. arXiv preprint arXiv:1511.08458, 2015.

[9] Federico Vaccaro, Marco Bertini, Tiberio Uricchio, Alberto Del Bimbo. Image Retrieval using Multi-scale CNN Features Pooling. arXiv preprint arXiv:2004.09695, 2020.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. arXiv preprint arXiv:1512.03385, 2015.

[11] Dor Bank, Noam Koenigstein, Raja Giryes. Autoencoders. arXiv preprint arXiv:2003.05991, 2020.

[12] Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556, 2014.

[13] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, Lihi Zelnik-Manor. ImageNet-21K Pretraining for the Masses. arXiv preprint arXiv:2104.10972, 2021.

[14] TensorFlow Hub. https://www.tfhub.dev.

[15] BiT Model on TensorFlow Hub. https://tfhub.dev/google/bit/m-r50x1/1.

[16] GitHub-Code. https://github.com/sayannath/Identical-Image-Retrieval.

[17] TripletSemiHardLoss. https://www.tensorflow.org/addons/api_docs/python/tfa/losses/TripletSemiHardLoss

[18] Kartik Chandra, Erik Meijer, Samantha Andow, Emilio Arroyo-Fang, Irene Dea, Johann George, Melissa Grueter, Basil Hosmer, Steffi Stumpos, Alanna Tempest, Shannon Yang. Gradient Descent: The Ultimate Optimizer. arXiv preprint arXiv:1909.13371, 2019.

[19] Arnulf Jentzen, Adrian Riekert. A proof of convergence for stochastic gradient descent in the training of artificial neural networks with ReLU activation for constant target functions. arXiv preprint arXiv:2104.00277, 2021.

[20] Stephan Wojtowytsch. Stochastic gradient descent with noise of machine learning type. Part II: Continuous time analysis. arXiv preprint arXiv:2106.02588, 2021.

[21] BigTransfer GitHub. https://www.github.com/google-research/big_transfer.

[22] Lev V. Utkin, Maxim S. Kovalev, and Ernest M. Kasimov. An explanation method for Siamese neural networks. arXiv preprint arXiv:1911.07702, 2019.

[23] Luca Bergamini, Yawei Ye, Oliver Scheel, Long Chen, Chih Hu, Luca Del Pero, Blazej Osinski, Hugo Grimmett, Peter Ondruska. SimNet: Learning Reactive Self-driving Simulations from Real-world Observations. arXiv preprint arXiv:2105.12332, 2021.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. arXiv preprint arXiv:2105.12332, 2017.

[25] Luis Perez, Jason Wang. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. arXiv preprint arXiv:1712.04621, 2017.

[26] Ilya Kostrikov, Denis Yarats, Rob Fergus. Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels. arXiv preprint arXiv:2004.13649, 2020.

[27] Marc Niethammer, Roland Kwitt, Francois-Xavier Vialard. Metric Learning for Image Registration. arXiv preprint arXiv:1904.09524, 2019.