Image Pair Matching: A Deep Learning Approach on the “Totally Looks Like” Dataset

COMP90086 Computer Vision Final Project

Lujie Ma
University of Melbourne
lujiem@student.unimelb.edu.au

Tao Yu
University of Melbourne
ty2@student.unimelb.edu.au

***Abstract*—In this paper, we delved into the Totally-Looks-Like challenge, a task inspired by human perception of image similarity. We measure the effectiveness of pretrained models and Siamese networks. Our results demonstrate that while pre-trained models showed promising results in this task, there are still indications of their ability to grasp abstract concepts is not superior. Siamese networks, are uniquely designed to compare the similarity of two images, but exhibit over-fitting concerns in this experiment, evidenced by differences in training and validation metrics. This study highlights complexity of image similarity prediction especially abstract features and emphasizes the need for continued exploration and optimization in this area.**

I. Introduction

Spurred by deep learning models and expansive data sets, computer vision has achieved remarkable success in tasks such as object classification and face recognition, often matching or even exceeding human performance. Yet, abstract image reasoning, where humans effortlessly discern similarities across images based on diverse factors like color, texture, or intangibles, remains elusive for machines. It requires the rich representation of image space in the brain, even recollect similar scenes stored in the memory to support the determination. [1]

The Totally-Looks-Like challenge tries to push algorithms to perceive image similarity like humans do. As shown in Figure 1, people sometimes see something and then stop for a moment and think, this looks exactly like something. Our experiment explores the possibilities of computer vision algorithms in this unique problem and try to find whether computer vision algorithms have a certain ability in finding the resemblances. We delved into different approaches and tested their proficiency in perceptual judgment of image similarity, including different pre-trained models and Siamese networks. Comparing the results tested under the provided data sets, it revealed that pre-trained models have certain abilities in finding the resemblances among different images. However, Siamese networks did not demonstrated comparable capabilities.

II. Related Works

A. Pre-trained Models

Pre-trained models are deep learning models that have been trained on large datasets in advance and saved for further use. They can be easily reused or further fine-tuned to adapt to specific tasks. Such models, particularly those trained on ImageNet, have already learned many general features like edges, textures, and colors. Hence, they can be leveraged as feature extractors to provide a solid starting point for various tasks. [2]

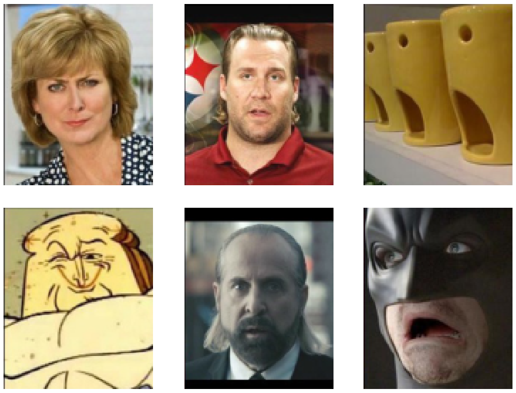

Fig. 1. Illustration of Totally-Looks-Like Challenge samples: for each column the top image exhibits visual similarities with the corresponding image below

Each of these pre-trained models possesses a distinct set of parameters, characterized by their size and complexity, necessitating a categorized discussion to understand their respective advantages and limitations. Table I provides a summary of the total number of parameters for various widely used pre-trained models.

Model Name	Total Parameters
resnet50	23,587,712
resnet101	42,658,176
resnet152	58,370,944
densenet201	18,321,984
vgg16	14,714,688
mobilenet	3,228,864
EfficientNetB0	4,049,571
mobilenetv2	2,257,984

TABLE I
Pre-trained Model Parameters

1) ResNet: is a deep neural network architecture introduced by Microsoft Research. Its innovative use of residual connections helps addressing the vanishing gradient problem in deep networks. ResNet has a very deep structure and achieved state-of-the-art performance. [3]

2) MobileNet: MobileNet is a lightweight deep neural network architecture proposed by Google. It uses depthwise seperable convolution to reduce the amount of calculation and model size. It has a much smaller number of parameters than many other deep network but still achieves comparable accuracy to other larger models on the ImageNet. [4] MobileNetV2 is a subsequent version of MobileNet, which has improved performance and efficiency.

3) DenseNet: DenseNet is shorted for Densely Connected Convolutional Networks, is distinguished by its unique architecture that incorporates short connections between layers, notably between layers situated close to the input and output. [5] The presence of these short connections is particularly beneficial for improving feature reuse, which is crucial for learning compact and accurate models, and for reducing the vanishing gradient problem, which helps to train deeper networks.

4) VGG16: The primary concept of VGG is to meticulously evaluate networks of incrementing depth employing an architecture with notably small (3x3) convolution filters. This demonstrates that by extending the depth to 16-19 weight layers, a substantial enhancement over previous configurations can be realized. [6]

5) EfficientNet: EfficientNet introduces a novel scaling approach that uniformly scales all dimensions—depth, width, and resolution—employing a straightforward yet highly effective compound coefficient. [7]

B. Siamese Networks

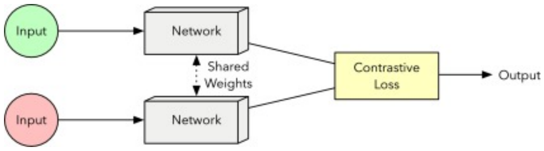

Fig. 2. The structure of the Siamese network [8]

Siamese networks, often called siamese networks, consist of a pair of neural networks that share their weights and aims at computing similarity functions. [8]

Owing to its unique network architecture (as depicted in Figure 2), the Siamese Network facilitates the simultaneous processing of two feature vectors, which are output by a shared-weight base network, thereby accepting them as inputs.

III. Method

A. Pre-trained Models

1) Data Preprocessing: According to the test candidates file, each left image has 20 candidates to compare with, however, in the train set there is only one matching right images for each image. Therefore, 19 dummy images will be randomly selected from all the right images except the matching images and then assign to the left images, making them 20 candidates. In this way, like the test file, we can compare the left image with the corresponding 20 right candidate images, then use the Top-2 accuracy test to determine the performances of the pre-trained model.

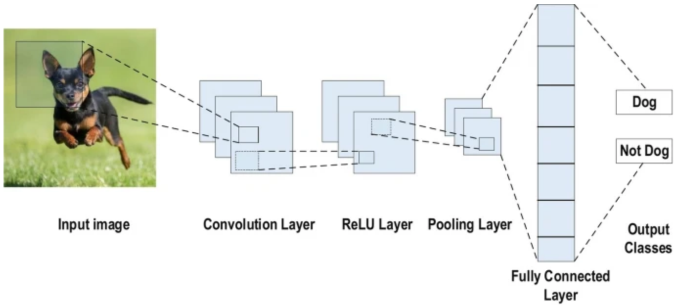

Fig. 3. An example of CNN architecture for image classification [9]

2) Feature Extraction: As Figure 2 shown, the CNN used for image classification contains two parts, layers before the fully connected layer extract and transform features from the original image, and the fully connected layer is responsible for classification. Once the fully connected layer of the pre-trained models is removed, the output will be the features extracted by the model. These feature representations extracted from the images will be vectors and ready for various tasks.

3) Evaluation: In the context of comparing image similarity, distance metrics are employed to measure the dissimilarity between feature vectors extracted by pre-trained models. Three common distance metrics utilized are the L1 distance (1), L2 distance (2), and Cosine distance (3).

$$d_{L1}(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{1}$$

$$d_{L2}(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2}$$

$$d_{\text{cosine}}(x,y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{3}$$

L1 and L2 measure how different two vectors are, and cosine similarity measures how similar two vectors are. Therefore, the larger L1 and L2 are, the more dissimilar they are, and the larger the cosine similarity is, the more similar they are. When comparing image similarity based on distance and utilizing Top-2 Accuracy, it's essential to note that similarity is the inverse of dissimilarity. Thus, it should be 1 - L1 for the L1 distance and 1 - L2 for the L2 distance.

After the similarities between each left images and right 20 candidates images came out, Top-2 accuracy will be applied to indicate the performance of different pre-trained models. The best model, including the type of pretrained model and the corresponding distance metrics, will be chosen to predict the test sets. For cosine similarity, it can be used directly.

B. Siamese Network

1) Data Prepossessing: To train a Siamese Network with constractive loss, we have to generate image pairs with label, and the format is [(left_img, right_img), label]. Fist, we need to generate matching pairs. According to the train.csv, put ground truth matching image pairs together and label them with 1. Then generate unmatching images pairs and label them with 0. After that, randomly shuffle their order, the first 80% is used as the training set, and the last 20% is used as the validation set, and then it can be used for training.

2) Training:

$$\mathcal{L}_{\text{contrastive}} = (1 - Y) \cdot \frac{1}{2} D^2 + Y \cdot \frac{1}{2} \max(0, m - D)^2 \quad (4)$$

Training a Siamese Network requires a suitable loss function that can handle pairs of instances. One common choice is the Contrastive Loss (equation 4). The goal is to make the distance between matching samples smaller than a certain threshold and make the distance between unmatching samples larger than the threshold.

During training, pairs of images and labels indicating whether they are similar or dissimilar are fed into the network. The network learns to map the input images to a feature space where the distance metric reflects their similarity.

C. Prediction

Since sigmoid is used as the activation function, once an image pair is fed to the Siamese network, the network will return a number between 0 and 1. This number reflects to what extent the model thinks the two images are similar. The closer to 1, the more similar it is; and the closer to 0, the less similar it is. We only need to feed the left image and the 20 right images one by one to the model and record the corresponding return values to complete the prediction.

IV. Results

A. Performance of Pretrained Models

Table II & Fig. 4. both demonstrate the Top-2 accuracy of different pre-trained models utilizing different distance metrics. Comparing these results, it shows that no matter what pretrained models were used, cosine similarity performed the best.

Comparing the performance of each pre-trained model as a feature extractor under cosine similarity, we can see that most models achieved more than 45% Top-2 accuracy rate. Among them, MobileNet performed best, followed closely by ResNet and DenseNet.

Model	Cosine	Euclidean	Manhattan
Resnet50	0.4600	0.4250	0.3300
Resnet101	0.4350	0.4050	0.3475
Resnet152	0.4650	0.4175	0.3500
Densenet201	0.4625	0.4100	0.3575
VGG16	0.4500	0.3475	0.2950
Mobilenet	**0.5050**[*]	0.4275	0.4025
EfficientNetB0	0.4550	0.3875	0.3800
MobilenetV2	0.3475	0.3300	0.3450

TABLE II
TOP-2 ACCURACY BY DIFFERENT PRE-TRAINED MODELS

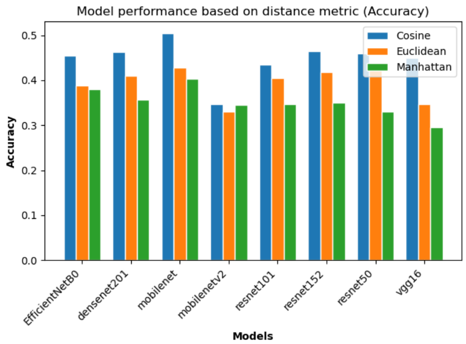

Fig. 4. Comparison of various model performances using different distance metrics: Cosine, Euclidean, and Manhattan.

B. Performance of Siamese Network

In our experiments, the Siamese network shows consistent progress on the training data: the loss value decreases from 0.6973 to 0.6484, while the accuracy improves from 0.5194 to 0.6272.These improvements reflect the gradual adaptation of the model to the training data.

However, the situation is slightly different for the validation data. The validation loss did not show a similar downward trend as the training loss, but instead increased in some cycles. Similarly, the validation accuracy did not show a significant increase with the training accuracy.

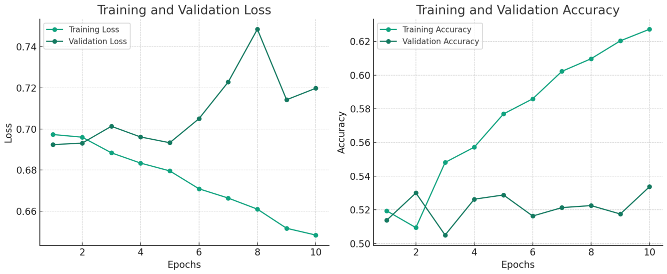

Fig. 5. The left graph shows the loss curve and the right graph shows the accuracy curve.

V. Discussion

A. Pre-trained Models

For pre-trained models, the number of parameters is often seen as an indicator of model complexity and computational cost. Theoretically, the more parameters, the better the model's representation, but it also increases the risk of overfitting. From our experiments, despite the fact that deep models such as ResNet152 have a large number of parameters, they do not significantly outperform lightweight models such as Mobilenet, which achieves the highest Top-2 accuracy at cosine similarity. This emphasises the importance of choosing the right model and similarity metric for task performance.

The effectiveness of prediction using pre-trained models was investigated by examining several sets of images that were accurately and inaccurately predicted using cosine similarity.

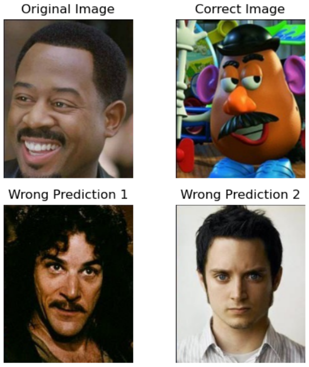

Fig. 6. Incorrect predictions using cosine similarity on features extracted by ResNet50.

Our experiments showed that this method led the model to identify precise similarities, rather than capturing the abstract similarities encapsulated in the term 'TOTALLY looks like'. For example, in one particular incorrect example (shown in the figure 6), the model identified two images as most similar to the original; although both were faces similar to the original, they lacked the nuanced shape similarity sought in this dataset. This observation suggests that the model may not be well suited to the unique requirements of this dataset.

In addition, different similarity metrics have a significant impact on model performance. While cosine similarity performs well in most models, certain models may show better performance in other similarity metric spaces. This emphasises the importance of choosing the appropriate similarity metric based on the particular task and dataset.

In the end, we achieved an accuracy of **0.534** on Kaggle by adopting the cosine distance and MobileNet.

B. Siamese Networks

When experimenting with siamese networks, we did not set 19 confounding prediction options for the validation set due to its particular structure and design goals. The core of the siamese network is to compare the similarity of two inputs, rather than directly predicting multiple categories for each input. Thus, for our task, it directly outputs a prediction of whether each pair of images is similar or not.

Based on this consideration, we decided to submit the model's predictions on the validation data directly to Kaggle. Through this unconventional approach, we tested the performance of the model, and its accuracy is **0.182** on Kaggle.

By closely observing our loss and accuracy curves (shown in the fig 5) during the training phase, it became clear that as the epochs progressed, the validation loss did not show signs of decreasing and the validation accuracy did not exhibit an upward trend, indicating a discrepancy between our training and validation metrics. This divergence is often a sign of overfitting, suggesting that the model may be over-optimised for specific patterns in the training data that do not necessarily apply to the validation data.

VI. Conclusion

Our exploration of image similarity prediction challenge under Totally-Looks-Like dataset using pretrained models and Siamese networks yielded interesting results.

Pre-trained models: The results show that multiple pre-trained models can achieve a certain accuracy on this task through cosine similarity. While deeper models with more parameters are traditionally considered more expressive, our experiments show that lightweight models like MobileNet perform better in this challenge. However, although we achieved 53.4% accuray on Kaggle, as our incorrect prediction example shown, the model's ability to understand abstract concepts needs further development.

Siamese Networks: The unique structure of Siamese networks is designed to evaluate the similarity between two inputs, which presents both opportunities and challenges. The poor performance of our model on Kaggle, particularly the observed differences between training and validation metrics, suggests potential overfitting. More research and experiments are needed.

VII. Future Work

In our experiments, although the pre-trained models and the Siamese network provided an interesting approach to the image similarity task, their performance did not meet our expectations. In view of this, future work will mainly focus on further optimising and improving the model to achieve better generalisation performance.

Exploring transfer learning: Given the challenges of the model in capturing the abstract similarity of the term "TOTALLY looks like", transfer learning could be a potential avenue to explore. Transfer learning is used to improve a learner from one domain by transferring information from a related domain. [10] By taking knowledge from a pre-trained model and applying it to our specific task, we hope to achieve better performance.

Improving Siamese Networks: Given that Siamese networks do not perform well on Kaggle, we believe that their structure and loss function may need to be tuned. In particular, the introduction of regularisation techniques, data augmentation or further tweaking of the network structure may help to improve their generalisation ability.

In conclusion, although this study provides valuable insights into image similarity prediction, much more can be done to further optimise and improve the performance of the models.

References

[1] Amir Rosenfeld, Markus D. Solbach, and John K. Tsotsos. Totally looks like - how humans compare, compared to machines, 2018.

[2] Rahul Gowtham Poola, Lahari Pl, and Siva Sankar Y. Covid-19 diagnosis: A comprehensive review of pre-trained deep learning models based on feature extraction algorithm. *Results in Engineering*, 18:101020, 2023.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[4] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[5] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[7] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[8] Gustavo H. de Rosa and João Paulo Papa. Chapter 7 - learning to weight similarity measures with siamese networks: a case study on optimum-path forestthe authors appreciate são paulo research foundation (fapesp) grants 2013/07375-0, 2014/12236-1, 2017/25908-6, 2018/15597-6, 2018/21934-5 and 2019/02205-5, and cnpq grants 307066/2017-7 and 427968/2018-6. In Alexandre Xavier Falcão and João Paulo Papa, editors, *Optimum-Path Forest*, pages 155–173. Academic Press, 2022.

[9] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), Mar 2021.

[10] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, 2016.