Totally-Looks-Like

Pangfeng Zheng

Email: pangfengz@student.unimelb.edu.au

Jiacheng Lu

Email: jiachengl9@student.unimelb.edu.au

***Abstract*—This study aims to develop an algorithm that identifies similar-looking image pairs. The data-set is a subset of the Totally-Looks-Like (TLL) data-set, comprises various image pairs that might resemble each other due to similar colors, shapes, textures, poses, or facial expressions. Participants are tasked with designing an algorithm that, given one image from a pair, identifies its match from a list of potential candidates. In this study, we develop experimental models based on CNN to try to reproduce the pairs by extracting features, we also try existing models and used to compare the performance. In our experiment, the performance of AI still has a large gap with human on this classification task. The testing result shows existing models are generally perform better than our own. We discuss and analyze these results, and provide suggestions for future direction on improvement.**

1. Introduction

Algorithms have made significant strides in Computer Vision, a good model often can provide high speed and high performance in tasks comparing with human. However, when it comes to a complex task such as abstract reasoning and flexible interpretation of images, a noticeable gap persists between machine and human performance. The 'Totally-Looks-Like Challenge' endeavors to bridge the existing gap by prompting participants to develop algorithms capable of identifying and matching similar-looking image pairs from a diverse data-set. This study delves into the intricacies of this challenge, try to explore and experiment various models to discern image similarities, critically analyzing results.

2. Literature Review

1) Amir Rosenfeld, Markus D. Solbach, and John K. Tsotsos, "Totally looks like-how humans compare, compared to machines" 2018 [1] - delve into the perceptual judgment of image similarity by humans. They introduce a new dataset "Totally-Looks-Like (TLL)," and conducted experiments to reproduce the pairings using generic and facial features extracted from state-of-the-art deep convolutional neural networks. They also carried out additional human experiments to verify the consistency of the collected data.The result found that machine-extracted representations performed poorly in reproducing the human-selected matches.
2) Olivier Risser-Maroix, Camille Kurtz, and Nicolas Loménie, "Learning an adaptation function to assess image visual similarities " 2021 [2] - introduce an innovative approach to address the challenge of assessing visual similarities between images. Utilizing various layers of a categorization-based CNN (pretrained on ImageNet) as an approximation of the visual cortex, and provide an adaptation function, which serves as an approximation of the primate IT cortex, within the metric learning framework. The result shows a significant improvement in retrieval scores on "Totally Looks Like" image dataset. The study provides a fresh perspective on the task of learning visual image similarities and offers a promising direction for future research in the domain of image similarity assessment.

3. Data-set

The data-set used in this experiment categorized as "left" and "right". These pairs have been further divided into 2,000 training pairs and 2,000 test pairs. For each "left" image in the test set, there are 20 potential "right" images, including the actual match and 19 randomly selected noise from the test set.The images have been sourced from the internet and have undergone automatic resizing and cropping to dimensions of 200 x 245 pixels. It's worth noting that the data-set may contain images that appear more than once, paired with different matches.These will be used as the foundation for the subsequent exploration and experimentation in this study.

4. Method

The methods we mainly use can be divided into two categories. One is to use the basic CNN model to binary differentiate images for classification tasks, and the other is to use existing models to extract the differences between feature values, thereby determine whether two pictures can be a pair.

4.1. Basic CNN Model

A basic Convolutional Neural Networks (CNN) model is created as a baseline. Our baseline model is also called Siamese neural network, which is different from ordinary neural networks that only receive one input. Siamese network is designed for comparing the similarity of two different inputs, it works together on two different inputs at the same time and uses the same weights to calculate a comparable output, by using a binary cross-entropy loss while training to determine whether two images match (1) or do not match (0) . The binary cross-entropy loss is written as:

$$L(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$$

- y is the true label (either 1 for a match or 0 for a mismatch).
- $\hat{y}$ is the predicted probability of the sample being a match.

4.1.1. Data Process. Based on there are no labels in the data set, we obtain "left" images and "right" images respectively from the data set containing the correct pairing, and mark the paired images as "1", indicating a match. To increase to the training data, the left image is paired with a different right image, which is marked as "0", indicating a mismatch. Positive and negative samples are combined to form the final training data set.

4.1.2. Model Structure.

- **Input Layer**: This layer receives an input image of shape (245, 200, 3).
- **Convolutional Layer 1**: Detect low-level features such as edges and textures.
- **Convolutional Layer 2**: Further refines the features detected by the previous layer.
- **MaxPooling Layer**: Reduces the spatial dimensions of the feature maps, retaining only the most salient features.
- **Flatten Layer**: Transforms the 2D feature maps into a 1D vector.
- **Dense Layer**: Acts as a classifier on the extracted features, produce a feature vector that can be used for comparison.

4.2. Other pre-trained model

Pre-trained models with different characteristics are used to compare the performance of different networks on this image matching task.

4.2.1. ResNet. We used ResNet-18, ResNet-50 and ResNet-152 respectively. Different from the basic CNN model, Residual network mainly introduces the concept of "Residual Block" and solves the gradient disappearance and gradient explosion problems in deep neural networks through the shortcut mechanism (shortcut connections) [5], [6], thus helping the model to better update the weight W during training. Assume the input is x, the function of the residual block can be expressed as:

$$F(x) = x + H(x) \tag{1}$$

- $H(x)$ is the output of the convolutional layer and other operations in the residual block.

4.2.2. DenseNet. The difference from ResNet is that output of DenseNet is a connection rather than a simple plus like ResNet. DenseNet-121,DenseNet-169 and DenseNet-201 are used in the study. Dense networks are mainly consisted of two parts: dense block and transition layer. The connection operation between input and output is achieved through dense connection to learn the direct mapping between input and output [7], [8]. The function can be written as:

$$x_l = H\left([x_0, x_1, \ldots, x_{l-1}]\right) \tag{2}$$

- $[x_0, x_1, \ldots, x_{l-1}]$ is the connection of the outputs of all previous layers.
- H are the operations of layer l (such as convolution, batch normalization, and activation functions).

4.2.3. AlexNet. As the first truly deep convolutional neural network, AlexNet is a milestone in the development of deep learning. By introducing Dropout and local response normalization (LRN) technology, the generalization ability of the model has been greatly improved. For LRN, it mainly enhances the larger response of the neuron by suppressing the response of neighboring neurons [4]. The calculation process can be expressed by the following formula:

$$b_{x,y}^{i} = \frac{a_{x,y}^{i}}{\left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a_{x,y}^{j})^2\right)^{\beta}} \tag{3}$$

- $a_{x,y}^{i}$ is the pixel value of the ith feature map at position (x, y).
- $b_{x,y}^{i}$ is the value after LRN.
- N is the total number of feature maps.
- n is the number of adjacent feature maps considered during normalization.
- k, α and β are hyper-parameters that control the degree of normalization.

4.2.4. VGG. The VGG-16 model mainly uses multiple consecutive 3×3 convolution kernels to extract features [3], building a network with large depth and width. Its initial layers capture low-level features like edges and textures, while deeper layers capture high-level semantic features.

4.3. Adaption Module

Olivier Risser-Maroix, Camille Kurtz, and Nicolas Loménie, "Learning an adaptation function to assess image visual similarities " 2021 [2] introduces a module called Adaption. The objective is to devise an adaptation mechanism that effectively narrows the gap between the "left"

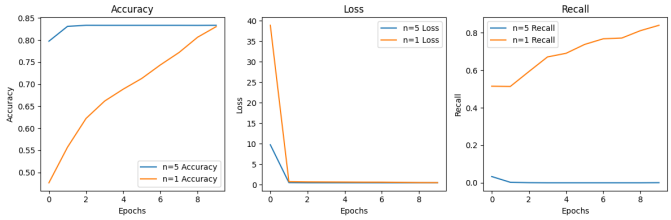

Figure 1. Accuracy, Loss and Recall for Baseline

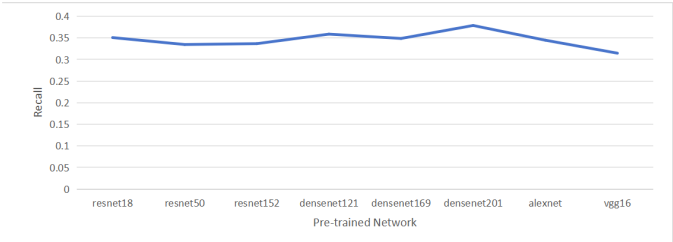

Figure 2. Recall Values for Different Pre-trained Networks

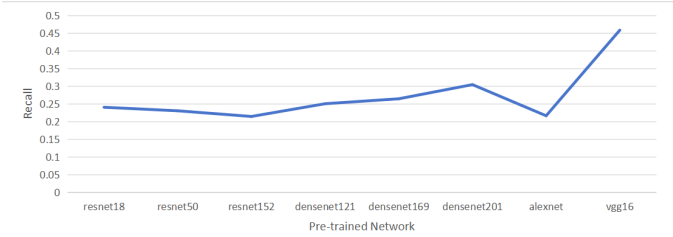

Figure 3. Recall Values for Different Pre-trained Networks After Adding Adaption Module

embeddings (dleft) and their corresponding "right" embeddings (dright), more so than for any other non-corresponding data points. The implementation idea is to train a Fully-Connected Neural Network. Using the trained Adaption module to extract more useful features from existing image embeddings. Then use cosine distance to evaluate the image similarities.

5. Experiment

In this section, we discuss the experiment process. The whole experiment will be separated into two ways, including different progresses of data processing and models' choice.

5.1. Data Process

We select the last 500 image pairs from the train dataset as the validation set. So the training data is 1 - 1500 image pairs, and validation data is 1501 - 2000 image pairs.

5.2. Baseline

First, we created a simple CNN model for training. Due to have only positive samples (correct image matches) may cause the model to be biased to predict all image pairs are matches because it has not learned mismatching samples, we introduce n negative samples (wrong pairings) for each left image to ensure that the model generalizes better to unseen data.Since our hardware equipment is not enough to support the operation of excessively large amounts of data, we only try $n = 1$ and $n = 5$ to increase training data. However, according to the test results shown as Figure 1, adding the number of negative samples seems like will lead the unbalanced of training data, and bring a negative influence to the model performance.

5.3. Pre-trained Model

We reproduced some of the pre-trained models discussed by Rosenfeld et. [1]. We used the pre-trained models such as resnet18, densenet121, etc. to extract features f_{pre}. The size of the train set is 2000, we used the last 499 images as the validation set. We randomly added 20 right images to each left image in the validation set, including a right image paired with the left image. Then used cosine distance to evaluate the image similarities. The shorter the distance, the more similar the two images are. We considered the successfully predicted image as TP(True Positive), the unsuccessfully predicted image as FN(False Negative). The equation for recall can be written as $\frac{TP}{TP+FN}$ The recall values for each model are shown in Figure 2.

5.4. Adding Adaption Module

After evaluating the performance of the pre-trained models, we trained an Adaption module by using f_{pre} for each of the pre-trained models. After completing the training of the Adaption module, we used the same way in section 5.2 to randomly add 20 right images to each left image in the validation set, including a right image paired with the left image. Then calculate the cosine distance. The recall values for each model are shown in Figure 3.

6. Analysis and Discussion

6.1. Baseline Model

According to the data shown in figure 1, we can see that for the self-created baseline model, as n (numbers of negative samples) increases, the model's accuracy will increase, but recall will decrease accordingly. The model is becoming biased towards predicting the main category (negative samples) and focusing on learning features that distinguish negative samples, does not capture the nuances of positive examples.

6.2. Pre-trained Model and Adaption Module

From Figure 2 and Figure 3, the recall values for each network are decreasing at the same time except for model

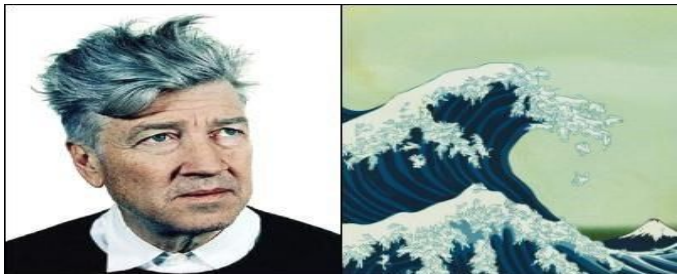

Figure 4. Example of Puzzling Paired Images

VGG16. The difference here is caused by the incorrect parameters while training the adaption module and the data set is not good enough. From the data set, we can see some unmeaningful paired images such as Figure 4. It is hard for the model to extract the specific features from these two images that can indicate that they can be paired. One is hair, and the other one is ocean waves. Although their colors are almost the same, the textures from these two images are almost not the same, one is human facial, the other one is ocean waves. In addition, the adaption module uses the cosine distance between two images to calculate the loss in order to decrease the distance between paired images. Because some paired images lack meaningful features, the adaption will not extract useful features all of the time.

As shown in Figure 3, model VGG16 performs best. The Figure 5 illustrates the incorrectly paired images and the correct paired images. We can see the first row of Figure 5, the original paired images which are the left and middle images in Figure 5 are all women, and the similarity between them is the shape of their hair, but the colors are not the same. From the predicted paired image which is the right image in Figure 5, the hair color, the shape of the eyes, and the shape of nostrils between the left and right images are almost the same. So the model predicted them as paired because of their similarities are more than the similarities between the left image and the right image. From the second row of Figure 5, the problem here is the same, although the similarities between the left image and the middle image are the color of their hair, the facial features are not the same. The right image's shape of hair and the shape of the chin are the same as the left image. So the left image and the right image are predicted to be paired. From the third row of Figure 5, the left image and the right image have human faces, it is easier for the model to predict them as a pair. However, the consideration for the original paired images is about their body color.

The human way to judge two images as a pair is not the same as our model. Humans will determine which feature is most important and which feature is not important. However, our model will compare the features extracted from two images equally and then calculate the similarities. So our model cannot get a good performance in this task. The improvement method is considering the importance of the features when calculating the similarities.

Figure 5. Example of Incorrectly Paired Images. The left and middle images are paired. The left and right images are predicted to be paired by our model.

7. Conclusion

In this report, we proposed a baseline model to test how to improve the model's ability of matching similar image. We believe reasonably increase negative samples will improve the performance, but how to get balance is still a challenge. We also researched and evaluated how to use cosine distance to indicate image similarities. Then use the adaption method to extract more useful features that can decrease the paired images' distance. However, the method of calculating similarities focuses on the generic features, not on some important specific features. As a result, the model will not perform generalization on the data set.

8. Future Direction

Try to determine the importance of the image features then calculate the similarities with the weights. Such as an attempt to divide images into different types of features and assign different weights for them.

Research other ways to make the adaption model more generalized for both the train set, validation set, and test set.

Acknowledgments

The author thanks Olivier Risser-Maroix, Camille Kurtz and Nicolas Lomenie for their effort on improvements to capture visual similarities between images. This work is based on project provided by University of Melbourne COMP90086.

References

[1] A. Rosenfeld, M. D. Solbach, and J. K. Tsotsos, *Totally looks like-how humans compare, compared to machines*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 1961–1964, 2018.

[2] O. Risser-Maroix, C. Kurtz, and N. Loménie, *Learning an adaptation function to assess image visual similarities*. 2021 IEEE International Conference on Image Processing (ICIP), pp. 2498–2502, 2021.

[3] ChengZiYa21, *Deep learning - detailed explanation of VGG16 model*. CSDN, 2022. [Online]. Available: https://blog.csdn.net/qq_42012782/article/details/123222042

[4] glq, *A review of classic convolutional neural networks: AlexNet*. ZhiHu, 2023. [Online]. Available: https://zhuanlan.zhihu.com/p/618545757?utm_id=0

[5] ChouXianYu, *ResNet50 network structure diagram and detailed structure explanation*. ZhiHu, 2023. [Online]. Available: https://zhuanlan.zhihu.com/p/353235794

[6] MathWorks, *ResNet-18 convolutional neural network*. MathWorks Documentation, 2023. [Online]. Available: https://ww2.mathworks.cn/help/deeplearning/ref/resnet18.html.

[7] MathWorks, *DenseNet-201 convolutional neural network*. MathWorks Documentation, 2023. [Online]. Available: https://ww2.mathworks.cn/help/deeplearning/ref/densenet201.html?s_tid=doc_ta.

[8] Culture and Technology Jun, *Densely Connected Network*. ZhiHu, 2023. [Online]. Available: https://zhuanlan.zhihu.com/p/619410626?utm_id=0