Facial Alignment and Lip/Eyes Color Modification

16/05/2024

KEISUKE TSUJIE

Abstraction

The investigation for tasks one and two of the module assignment is discussed in this report. The first challenge is to develop a system that uses face alignment. Convolutional neural networks, or CNNs, are the method used in the paper. Consequently, a model that is strong only in front-facing images for different genders and ethnicities is developed. The second challenge is modifying the color of lip and eyes. The method used for this work is to index the points on an aligned image and determine the location to modify the color. Assuming that the alignment is accurate, this method works well.

Table of Contents

Task One: Face Alignment

1 Introduction

The task given is face alignment of an image data. In this practice, a NumPy formatted image data is given, and it is required to detect the feature points of faces.

Figure 1: The original picture (left) and the picture after face alignment

The approach taken here involves utilizing Convolutional Neural Network (CNN) to recognize the key features of the face. CNN is a neural network which consists with convolutional layer, pooling layer, and the fully connected layer. Throughout each layer, it up samples the feature in the image, reduces the spatial misalignment, and integrate the image to create output. These methods taking an advantage of the robustness of CNN to the various images such as rotated or non-centred data. Later in this essay, we will discuss the methods employed for the experiment, and having further analysis to the result of the experiment.

2 Outline of methods employed

The main target of the procedure is to train a CNN model. To train a model, the following sequence is taken:

Figure 2: Flowchart about process brief

We focus on a few key components that significantly affect the final model's performance for each step.

2.1 Setting up dataset

The data configuration phase consists of two functions: preprocessing and loading data. The preprocessing of the dataset is the factor that has the biggest impact on the model's performance in this phase. This experiment's code was written with reference to Vinay Kalmoodkar's "Facial-Landmark-Detection" project.

Figure3: Flowchart about data preprocessing

Reducing unnecessary information is the goal of data re-sizing in order to facilitate feature identification. Normalization and grayscaling were used to approach similar targets. They enhance the CNN model's performance by simplifying the image to reduce the computational load. Tensors will make it possible to batch the images in small sizes to enhance the model's functionality and help with the load of training data.

Figure 4: From the left, original image and after resizing/grayscaling/normalizing.

2.2 Configuration of Convolutional Neural Network

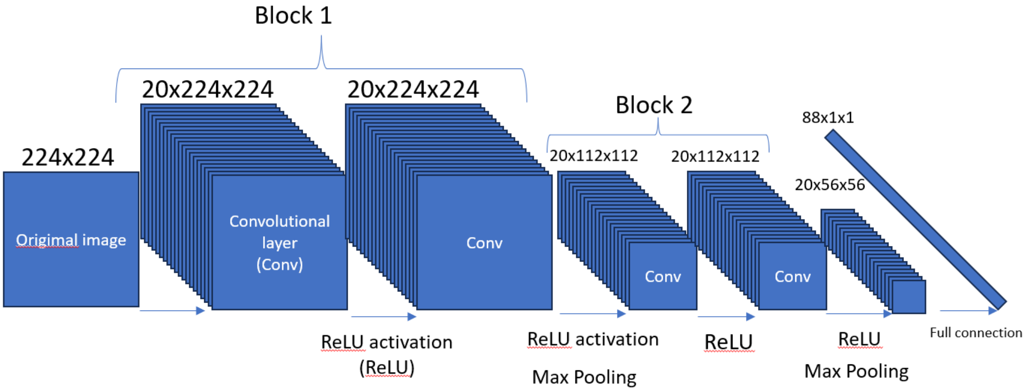

Figure 5: The architecture of the CNN model

"03. PyTorch Computer Vision" from "pytorch-deep-learning." Shows nice CNN architecture Two convolutional layers and one max pooling layer make up a single process block. Once the image data was sent to each block, it shrank in size by half. Finally, the full connected layer outputs an 88x1x1 dimensional key point value.

2.2.1 Investigation on activation function and structure of convolutional layer

To construct model, we concentrate on investigating the number of nodes in each convolutional layer and the choice of the activation function. Numerous nodes could be a trade-off between better performance and overfitting. We decided the number as 20 after trials of models with varying number of nodes. For the activation function,

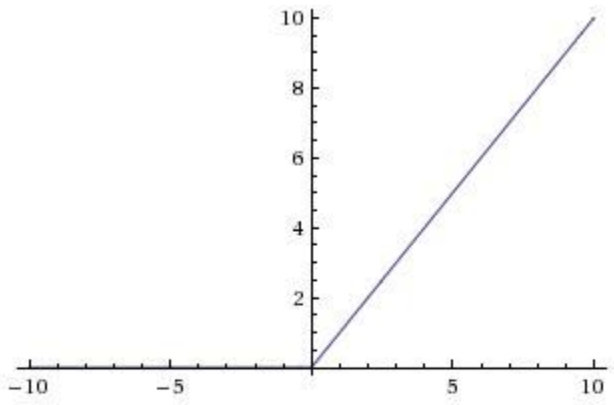

Figure 7: Graph of ReLU function [Danb, 2018]

ReLU is used to take an advantage of accelerates the convergence and helps cope with gradient death.

2.3 Model training process

We will now talk about the training procedure.

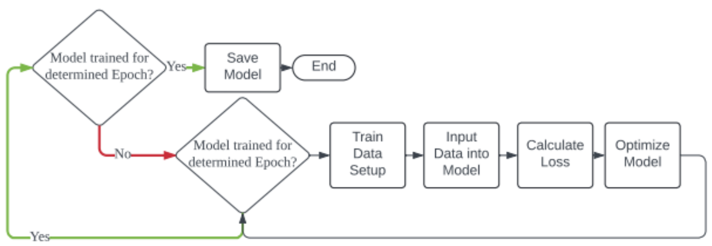

Figure 7: Flowchart of training loop

The flow chart explains the train loop over the epoch. In each iteration, a batch with a size of 10 will be given to model to create an output. For the loss calculation function, mean square error (MSE) is applied.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (\hat{y_i} - y_i)^2$$

It is expected to emphasize the locational difference in the output points and the actual points. Adam is utilized as the activation function. Adam can mitigate oscillations in learning by calculating moving average and adjusting learning rate.

$$opimized\ weight: W_{t+1} = w_t - \alpha \nabla_w L_{(w)}$$

3 Analysis of the performance of the trained model

Both quantitative and qualitative comparisons are included to analyze and explain the model's performance. To examine quantitative comparisons, the model's error against the training data is compiled into a cumulative error distribution. We will talk about the model's success and failure scenarios when comparing its output to the unknown data to make a qualitative comparison. For the second step of qualitative comparison, we'll suggest a few possible fixes to deal with the failure circumstances.

3.1 Quantitative analysis

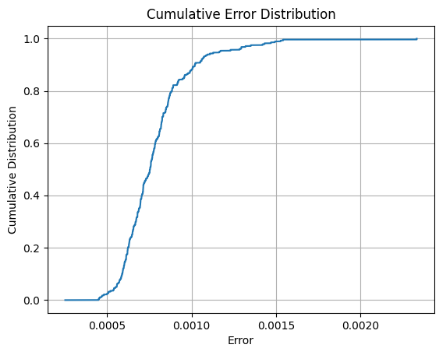

Figure 8: Graph shows cumulative error distribution of the error of model against training dataset.

For qualitative comparison, train data is inputted into the model. MSE loss function is used to keep the integrity between training. The collected data is compiled in cumulative error distribution. According to the data, in about 90% of the cases, the error value is in the range of 0.0005 and 0.0015 which means high accuracy of the model. However, since the high performance suggests the possibility of overfitting. It might cause difficulty for the model to prediction unseen data.

3.2 Qualitative analysis

In this section, we will examine the common elements across all cases to determine the inclination of effectively processed images, and we will also talk about the reasons behind the success of these elements.

Figure 9: Success cases

The model successfully aligns points with different genders, nationalities. Common feature is that the images include front-faced person with slight facial expression, which makes it easy recognizing the features. This could ensure that the model processing ease, producing the intended results. It may be worthwhile to note that the train data may contain photos with certain shared characteristics, making it easier for the model to provide the intended results.

Figure10: Failure cases

These failure cases show that the model has weaknesses with the cases include abnormal facial expressions and camera angles. For example, the left-hand image is taken from the lower angle, and the person opening the mouth confuses the model with mouth and nose alignment. The person in the right-hand image is tilted which makes some off-points on contour alignment. The model could be overfitted due to the performance is significantly lower in these cases.

3.3 Proposing potential solutions

To address these failures, we will propose two solutions. The first approach is to prevent overfitting. AdamW instead of Adam can be used as optimization function in training process. It contains L2 regularization in the weight decay which makes it possible to distribute weights evenly. Data augmentation (e.g. flipping or rotating the image) is also effective to make model experience more various data. The second approach is to change the architecture of the model. The model with this change could recognize more complex image data such as the failure case 1.

4 Conclusion

To solve the face alignment task, the approach to train CNN model was applied. As a result, the model performs significantly high performance in the training data. However, in unseen data, the model only performs well in the front-faced picture. It could be suggested that the CNN model is in overfitting states.

Task Two: Lip/Eyes color modification

1 Introduction

The second task is color modification of eyes and a lip of a face in an image. It requires to segment certain areas within the face which will be modified.

2 Methodology Explanation

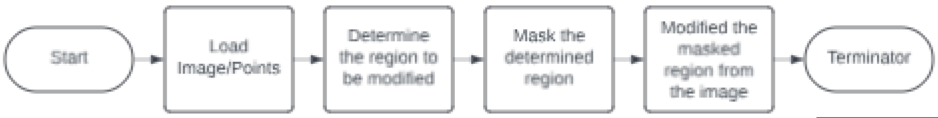

Figure 11: Flowcharts of methodology

The assignment brief had a facial photograph with a series of indexes on the face, which inspired my theory. These indexes are very helpful since they cover every area of the face with a variety of metrics, making it easy to extract the face features such as lips and eyes. This allowed for the creation of a mask to cover the testing image and apply color to the regions where the appropriate indexes appeared.

3 Example results and failure cases

Figure 12: Success and failure case

It demonstrates how the lips and eyes have been accurately adjusted to red. However, this approach is inappropriate for photos that lack accurate alignment points. It could be able to resolve this issue by developing a CNN model with the specific ability to identify lips and eyes to anticipate more accurate masking.

4 Conclusion

This method successfully modifies the area of the eyes and the lip of image data. Although, the performance depends on the accuracy of the face alignment.

Reference

Danb, 2018. *Kaggle.* [Online]
Available at: https://www.kaggle.com/code/dansbecker/rectified-linear-units-relu-in-deep-learning

Danb, 2018. *Rectified Linear Units (ReLU) in Deep Learning.* [Online]
Available at: https://www.kaggle.com/code/dansbecker/rectified-linear-units-relu-in-deep-learning
[Accessed 15 5 2024].

Kingma, D. P., 2014. *Adam: A Method for Stochastic Optimization,* s.l.: CoRR.

kpzhang93, 2017. *Gitub.* [Online]
Available at: https://www.zotero.org/gomen_asobase/collections/S9CMTS3K/items/VA4U3IZH/item-details

mrdbourke, 2023. *GitHub.* [Online]
Available at: https://github.com/mrdbourke/pytorch-deep-learning/blob/main/03_pytorch_computer_vision.ipynb

Yamashita Rikiya, N. M. D. R. K. G. T. K., 2018. *Convolutional neural networks: an overview and application in radiology,* s.l.: Insights Imaging.