CSC420 Project-Kpop Star Face Recognition and Emotion Detection

Mingdi Xie, utorid:xieming8

1. Introduction

In South Korean, due to the popularity of plastic surgery and their sophisticated makeup skills, people might find it hard to differentiate faces of celebrities. Moreover, fans of the Korean pop stars might want to know how their favourite stars are feeling when they are standing on the stage or posting a story on social media. In a concert, there are many fans raising up their phone and want to capture a nice photo of their favourite star, however it is hard to take one since human cannot react fast enough to capture a swift moment. Given the application of deep learning to the domain of facial recognition that has been widely used today. Me and my teammate decided to tackle this problem and satisfied the need of the fans by using computer vision technique and deep learning method. My teammate mainly focus on object detection to detect faces and then using face recognition to tell the name of the pop face. I mainly focus on emotion detection by implementing deep learning structure such as Convolutional Neural Network and generating portrait mode by using image processing technique such as Gaussian blur. Facial expression recognition has always been an active research area, and it is still difficult due to the high intra-class variation. Using traditional approaches such as SIFT, followed by a classifier trained on a database of images or videos works reasonably fine on images captured in a controlled condition, but fail to perform well on image with variation and partial faces.[1]Therefore, deep learning method is used here. The result doesn't expected to be perfect since some of the pop stars look almost the same and some facial expression could be shedding tears because of happiness. What is expected is the model could achieve good result on the ones that isn't vague to human.

2. Methods

2.1. Face detection and face recognition

For face detection both Multi-task Cascaded Convolutional Networks and Haar Cascade classifier is used by importing from library and the result is compared. For face recognition, a python library called "Google Images Download" is used to download the photo of the Korean pop stars then MTCCN is used to crop the face of the celebrities. CNN is built based on a paper "very deep convolutional networks"[2]. Images are normalized before passing in and 11 weight layers are trained in these settings. Next, we fine-tune our own dataset on Resnet[4] which is pre-trained network on VGG-face2. (Credit to my teammate)

2.2. Dataset

For emotion classification, the emotions are broken into 7 classes which are 'anger', 'disgust', 'fear', 'happy', 'sad', 'surprised' and 'neutral'. Then different model is implemented to improved the classification. The face emotion dataset is obtain from Aclab [3]. The data is a set of 32 $\times$ 32 grayscale face images. The faces have been rotated, scaled and aligned to make the task easier. Each of the face has been labeled and fallen into one of seven emotion categories. The dataset has already be spread into 3374 training images and 419 validation images and 385 test images. In figure 1 some of the data are illustrated. The training images are normalized before passing it into the model. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values.

(a) Anger (b) Disgust (c) Fear

fig 1: Some facial expression for anger, digust and fear

2.3. Fully Connected Neural network

First, I implemented a fully connected neural network. The neural network has only 2 hidden layers, with 32 neurons in the first layer and 64 neurons in the second layer. The output layer has 7 neurons and each indicate a score for the seven emotions. Softmax function is used at the end to give a clear percentage of each emotions. The loss function is Cross Entropy Loss, it measures the performance of a classification model whose output is a probability value between 0 and 1. The learning rate is 0.001, it is set small is to prevent vanishing gradient. The epouch is set to 200 and batch size is 16. Stochastic Gradient Descent is used as the optimizer and momentum is set to be 0.9. After obtaining the accuracy from the test set, I doubled the size of the neuron in each layer with 32 and 64 neurons respectively, and used dropout to reduce overfitting in neural net by preventing complex co-adaptations on training data. The table from figure 4 have shown the result of the final output.

2.4. Convolutional Neural Network with residual connection

Secondly, I implemented a Convolutional Neural Network with 2 convolutional layer and 2 max pooling with batch normalizing in between. The loss function is Cross Entropy Loss, learning rate is 0.001, epouch is 30, the optimizer is Stochastic Gradient Descent which converge faster than batch training because it performs updates more frequently and the momentum is set to be 0.9. The architecture of the network is shown in figure 2. After testing the resulting, the weight of the convolution filter bring up my curiosity. So I extract the weight and visualize it as a 2d image, the interesting result is in figure 5. Next I implement CNN with deeper network with 3 convolution layer and 3 max pooling layer with batch normalizing in between. The parameters are same as the previous network with 2 convolution layers and 2 max pooling in order to test out the effect after deepening the network. Next, I added a residual connection from the first convolution layer to the third convolution layer and compare the one without residual connection. All the other parameters are controlled to be the same to check the effect of residual block. This idea was used in ResNet[4] architecture for the training of 150 layers deep network [4]. The architecture is shown in figure 3.

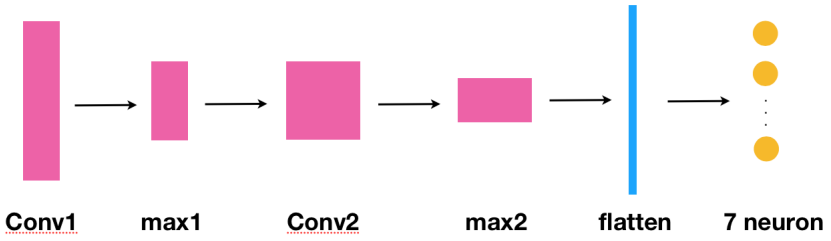

fig 2: CNN with 2 convolutional layer and 2 max pooling layer

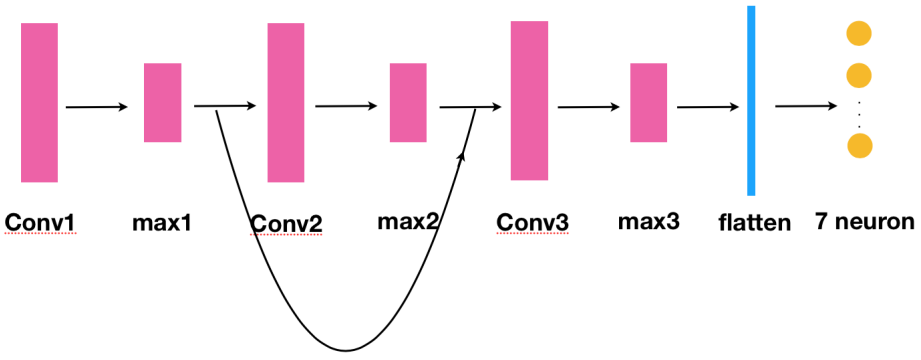

fig 3: CNN with 3 convolutional layer and 3 max pooling and residual connection

2.5. Canny Edge Image as input

Since peoples emotion expression can be mainly detected by the edges on the face. By applying canny edge on the input image before training might give better results since it is removing non-relevant information of the face such as the color of the skin. Moreover, I combine the original image with the canny edge image resulting in a 2 channel image as input and test it on the previous model with 2 convolutional layers and 2 max pooling.

2.6. Portrait mode

Lastly, I implement a portrait mode on the picture for the Korean pop star. Pre-trained U-net weight is used here from Thuyngch[5]. The Unet is trained for human segmentation. Next, the Korean pop star image is used as the input to obtain a binary mask from the Unet model. Since the binary mask have a property of ones and zeros, I can used that to differentiate the human body on the original picture. Subsequently Gaussian blur is used on the entire picture of the original image, then the blur image pixels is replaced by the original image based on the binary mask. One example is illustrated in figure 7.

3. Result and discussion

3.1. Face detection and face recognition

In comparison of Multi-task Cascaded Convolutional Networks and Haar Cascade classifie, MTCNN usually can detect faces that are hard to detect but also generates false positives, but in our case more faces is desired so false positives don't really matter. For face recognition, the 11 weight layers network achieve best validate accuracy around 0.7. Deepening network and increase dataset will make the model a little bit better but it is very limited because this network is for general image recognition instead of facial recognition. After fine-tuning on Resnet[4] the validation accuracy can hit as high as 0.75. The following figure shows the result of the output image. (Credit to my teammate)

fig: The result of face recognition from fine-tuning on Resnet

3.2. Fully Connected Neural network Result

Before testing out the result on the fully connected neuron network. The expected classification rate of the fully connected neuron network should be lower than convolution neural network since convolution neural network could obtain convolved feature map by sliding filter matrix. For the neuron network of 16 neurons and 32 neurons, I use a epouch of 200, the training loss decreases sharply after a few iterations and converge to around 2500 and the validation loss converge to around 50. Stochastic Gradient Descent is used as the optimizer since it converge faster than batch training because it performs updates more frequently. We can get away with this because the data often contains redundant information, so the gradient can be reasonably approximated without using the full dataset. The model achieve 69.45% of accuracy on the test set. After deepening the network and varying the dropout rate, the result is shown in figure 4. We can see from the table that as the neuron increases the test accuracy increases and dropout rate does have a effect on the performance with dropout rate equal to 0.2 having the best performance in both model. Dropout is an approach to regularization in neural networks which helps reducing interdependent learning amongst the neurons.

First layer neuron	second layer neuron	Dropout	test accuarcy
16	32	0	63.5%
16	32	0.2	66.8%
16	32	0.5	56.4%
64	128	0	71.2%
64	128	0.2	73.3%
64	128	0.5	69.9%

fig 4: The accuracy on the test set using different neurons and dropout rate

3.3. CNN Result

Next, the test is performed on CNN with 2 convolution layers and 2 max pooling. It acheive a accuracy of 78.04% which is 8.59% higher than the 2 hidden layers with 16, 32 neurons and 4.74% higher than the 2 hidden layer with 32, 64 neurons. Then I extract the filter weight of the first and second layer and visualize the result in figure 5. The interesting fact is that the second layer tempted to be darker and had less facial features comparing to the first layer. Then I test the result on the CNN model that has 3 convolutional layer and 3 max pooling. Figure 6 illustrate the result. From the table we can see the model achieve better accuracy than the model with only 2 convolutional layer and max pooling. Controlling the dropout rate, the accuracy of the model with residual connection is 1.7% higher than the ones without residual connection on average. Residual blocks are a variant of cross-channel connectivity, which smoothen learning by regularizing the flow of information across blocks. Residual allow gradients to flow through a network directly, without passing through non-linear activation functions. Non-linear activation functions could cause the gradients to explode or vanish and therefore bringing better result.

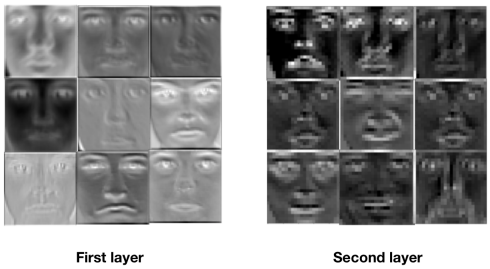

fig 5: Visualization of the filter weight

Structure	Residual	Dropout	test accuarcy
3 conv + 3 max	No	0.2	81.6%
	No	0.3	84.7%
	No	0.5	80.1%
	Yes	0.2	82.7%
	Yes	0.3	85.2%
	Yes	0.5	83.6%

fig 6: The accuracy on the test set using residual or without

3.4. Canny Edge Result

By applying canny edge on the input image before training is expected to give better results. However the loss become nan after a few iterations. After analyzing it, I found that cross entropy loss function is taking the log of 0, the reason could be too many features lost after applying canny edge. So I combine the original image with the canny edge image resulting in a 2 channel image as input. However, the accuracy turn out to be lower with only 49.88%. The reason could be the canny edge image have include some noise that affect the original image.

3.5. Portrait mode Result

The performance of the UNet turn out to be very well in predicting human segmentation. Due to lack of dataset, the loss cannot be calculated for the model, but based on the input image that is passed in, the performance seem to be very outstanding. For the portrait mode, a example in figure 7 is presented.

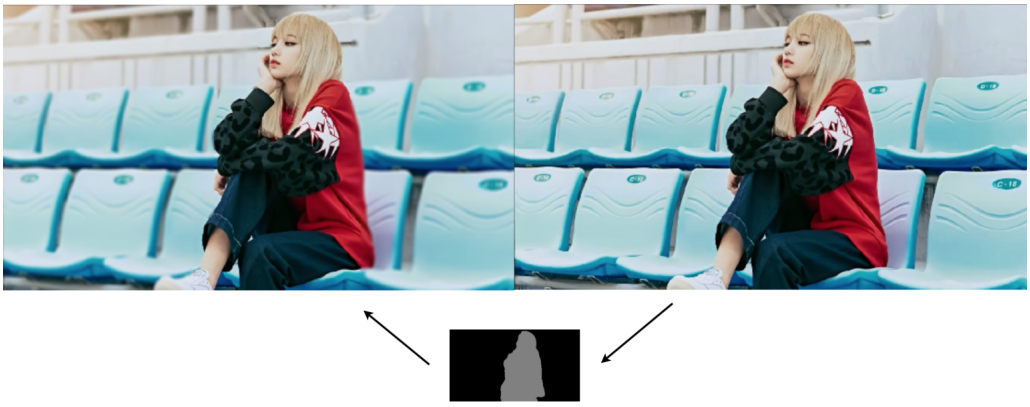

fig 7: left image is the portrait and right image is the origin

4. Main challenges

For CNN, the loss sometimes becomes nan which is a model divergence. Once the loss becomes nan after a certain pass, the model gets corrupted after backpropagating. In the model, the loss function is cross entropy, which is taking a softmax of a one-seven hot vector and then taking log of it , therefore issue such as -infinite will occur since if it is used on a very small positive that is closed to zero. In order to avoid this issue, I used a smaller learning rate and normalized the input before passing it in. Another challenges is labeling bias. The labels that used in the paper are one-hot-vector, however human could have multiple expression at the same time. For instance, fear and anger at the same time or happy and surprise. In another way, one expression might highly correlate with another expression. Human could interpret other feelings differently, therefore, the labels are very difficult to be accurate. Another main challenges will be different race might have different facial features, resulting in a different facial expression. Even though most of facial expression are generally the same, the subtle difference will be detect by the machine learning model and subsequently introducing variety or outliers. Therefore the best is the training data equally distributed on all race, or selected particular on one race and test on the same. Like Asian in our case or Korean will be even better. Gender might have a slight differences on the facial features as well. Since this paper is perform on a girls' Korean pop group, but the training data is combine from both males and females. Different face features due to gender differences could also influence the performance of the model. Given the challenges above and time constraint, I manually removed some of the different gender and race faces from the training data and manipulate some of the labels.

5. Conclusion and future work

Multi-task Cascaded Convolutional Networks performs quite well in face detection. Given a large training dataset the impact of the number of weight layers is not that significant. Resnet[4] could be used for face recognition and by fine-tuning it gives more desirable result. In this report, for emotion detection Convolutional neuron network perform significantly better than fully connected neuron network. Image after applying canny edge lost too many features and doesn't give promising result for emotion detection. In our model, CNN with residual connection outperform the one without residual connections. UNet and Gaussian blur could be applied to generate portrait for human or even other animals, therefore by using this model it can save a lot of time for image processing like manually cropping out the human. For future work, a real time emotion detector can be can be used on a video or even a using camera, so it can auto detect someone's emotions and then automatically capture that specific moment. And with the auto portrait mode, all the spectacular moment will never be miss on the camera.

References

[1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *Deep-Emotion: Facial Expression Recognition Using Attentional Convolutional Network*. University of California, Riverside, 2019.

[2] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *Very Deep Convolutional Networks For Large-scale Image Recognition*. University of Oxford, 2015.

[3] Toronto Faces Dataset. Retrieved from http://aclab.ca/users/josh/TFD.html

[4] K. He, X. Zhang, S. Ren, and J. Sun *Deep Residual Learning for Image Recognition, Multimed*. Tools Appl., vol. 77, no. 9, pp. 1043710453, Dec. 2015.

[5] Human-Segmentation. Retrieved from https://github.com/thuyngch/Human-Segmentation-PyTorchbenchmark