

Ranking Comedians Face Using CNN

Yuri lee
Department of Data Technology
Myongji University
Seoul, South Korea
E-mail: leyuri97@gmail.com
github:https://github.com/leyuri/Rankin_g-Comedians-Face

Abstract— 이 문서는 CNN 을 활용하여 코미디언 얼굴 순위를 매기는 연구에 관한 것이다. 우선 SCUT-FBP5500 Dataset 을 사용하여 얼굴 순위를 매기는 CNN 모델을 만든다. 그 과정에서 SCUT-FBP5500 Dataset 의 90%를 test set, 나머지 10%를 validation set 으로 사용한다. 그 과정에서 layer 추가, imageDataGenerator 과 dropout 등의 기법을 사용하여 val_loss 를 최소화 하는 모델을 찾아 성능을 올린다. 최종적으로 코미디언들의 이미지를 새로운 test set 로 사용하여 코미디언들의 외모 순위를 측정해본다.

Keywords—SCUT-FBP, Classification, Image, Ranking, CNN, ImageDataGenerator, ImageClassification, Face

I. INTRODUCTION

AI 가 사람의 외모까지 측정하고 평가할 수 있다면 어떨 것 같은가? 너의 얼굴을 평가하는 기계를 상상해봐라. 너의 얼굴을 보고 점수를 매긴다면 어떤 기분이 들 것 같은가? 기분이 좋을 것 같은가? 혹은 나쁠 것 같은가? 그와 동시에 나는 한국의 코미디 쇼인 무한도전에서 ‘미남이시네요’라는 주제로 프로그램을 진행했던 것이 떠올랐다.

당시 그 프로그램의 기획 의도는 코미디언 멤버의 외모 논쟁을 끝내기 위함이었다. 투표를 위하여 국내에서 현장 투표, 인터넷 투표, 성형외과 전문의 투표를 진행하였다. 그 외에도 12 개국 15 개 도시에서 3,337 명이 참여한 가운데 외국인 투표도 진행하였다. 이 모든 투표를 합친 최종 결과로 노홍철-유재석-하하-길-정형돈-정준하-박명수의 순서대로 순위가 매겨졌다. 따라서 나는 이 코미디언의 얼굴을 딥러닝을 통해 모델에 적용해본다면 흥미로울 것 같다고 생각했다.

본 연구에서는 미리 추출된 Dataset 을 활용할 것이다. 이 Data set 은 이미지와 함께 Beauty score 가 1~5 사이로 매겨져 있다. Beauty score 는 중국 60 명의 자원봉사자가 이미지를 보고 점수를 매긴 결과이다. 우선적으로 이 Dataset 을 통해 외모 점수를 평가하는 모델을 만들 것이다. Full Connected Layer 를 통해 이미지를 학습시킨다고 가정해보자. 이미지 size 를 조금만 더 크게 해도 파라미터 수가 기하급수적으로 늘어날 것이다. 계산해야 할 파라미터도 많지만, 이를 미분한 다음에 다시 학습 시켜야 하는 것도 많다. 이미지 데이터는 학습 과정에서 dimension 이 매우 높아지기 때문에 제대로 모델을 학습시키는 것은 쉽지 않다. 따라서 Convolution 으로 부터 도출된 Convolution layer 를 활용할 것이다. 각종 레이어 및 파라미터가 다른 8 개의 케이스들을 살펴볼 것이다. 그 8 가지 케이스 안에서는 여러 파라미터를 변경하여 어떻게 조절해야 알맞은 모델 방향으로 나아가는지 볼 것이다. 각각의 케이스 모델들을 val_loss 를 plot 으로

그려보고 확인하고 모델을 만들어 예측 점수와 실제 점수를 확인한다. 이후 코미디언 얼굴들을 test dataset 으로 사용하여 만들어진 모델로 그들의 얼굴 점수를 얻고 순위를 매겨본다. 최종적으로 한국의 코미디 쇼인 무한도전에서 최종 투표 결과와 비교해본다.

II. RELATED WORK

Beauty_Score[1] - 이 연구에서는 SCUT-FBP5500 데이터 세트를 사용해 CNN 모델을 사용하여 모델을 만들었다. 그 이후에 BTS 이미지를 사용하여 BTS 의 얼굴 순위를 지정했다. 연구에서 가장 손실이 적은 모델은 0.3863 이었고 모양 순위는 순서대로 정국 – V – 진&RM – JHope – 지민 – 슈가로 측정되었다.

III. DATA DESCRIPTION

A. Train set

SCUT-FBP5500 데이터 세트는 2000 명의 아시아 여성 (AF), 2000 명의 아시아 남성 (AM), 750 명의 백인 여성 (CF) 및 750 명의 백인 남성 (CM)을 포함하여 인종과 성별이 다른 4 가지 하위 세트로 나눌 수 있다. SCUT-FBP5500 의 대부분의 이미지는 인터넷에서 수집되었으며, 아시아 얼굴의 일부는 DataTang, GuangZhouXiangSu 및 실험실에서, 일부 백인 얼굴은 10k US Adult Faces 데이터베이스에서 가져왔다. 모든 이미지는 [1, 5]에서 총 60 명의 자원 봉사자에 의해 Beauty score 가 표시되며 86 개의 얼굴 랜드 마크가 각 이미지의 중요한 얼굴 구성 요소에 위치한다.

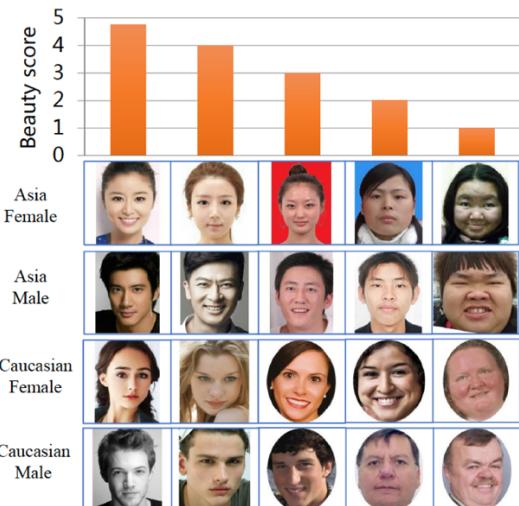


Fig. 1. The images with different facial properties and beauty scores from the proposed SCUT-FBP5500 benchmark dataset. [2]

B. Test set

코미디언 test dataset 은 다음과 같이 7 명의 이미지를 사용한다. 해당 이미지는 대한민국의 무한도전이라는 프로그램에서 ‘미남이시네요’라는 방송 중 외모 평가 투표를 위해 코미디언들이 찍은 사진이다.



Fig. 2. The images of Infinite Challenge Members

IV. EXPERIMENTS: INSTANCE SEGMENTATION

A. How to collect data

Train 데이터 수집은 사이트[3]에서 다운받을 것이다. SCUT-FBP5500 에서 제공하는 데이터셋을 활용한다.

Test data 수집은 방송국 사이트[4]에 들어가서 이미지를 다운 받았다.

B. Method of performance comparison

val_loss 값을 통해 모델의 성능을 비교할 것이다. 따라서 val_loss 를 최소화 시키는 방향으로 training 이 진행 될 것이다. 보통 Early stopping 객체에 의해 트레이닝이 중지되었을 때, 그 상태는 이전 모델에 비해 일반적으로 validation error 가 높은 상태가 된다. Early stopping 을 하는 것은 특정 시점에 모델의 트레이닝을 멈춤으로써, 모델의 validation error 가 더 이상 낮아지지 않도록 조절할 수는 있다. 하지만 중지된 상태가 최고의 모델은 아닐 것이다. 따라서 가장 validation performance 가 좋은 모델을 저장하는 것이 필요한데, keras 에서는 이를 위해 ModelCheckpoint 라고 하는 객체가 존재한다. 이 객체는 validation error 를 모니터링하면서, 이전 epoch 에 비해 validation performance 가 좋은 경우, 무조건 이 때의 parameter 들을 저장한다. 이를 통해 트레이닝이 중지되었을 때, 가장 validation performance 가 높았던 모델을 반환할 수 있다.

```
from keras.callbacks import ModelCheckpoint
mc = ModelCheckpoint('best_model.h5',
                      monitor='val_loss', mode='min',
                      save_best_only=True)
```

위 ModelCheckpoint instance 를 callbacks 파라미터에 넣어줌으로써, 가장 validation performance 가 좋았던 모델을 저장할 수 있게 된다. 즉 가장 val_loss 가 낮은 model 을 얻어낼 수 있다.

C. Case 8 (The cases have been modified with reference to [1])

하이퍼 파라미터를 다르게 한 8 개 케이스들을 살펴볼 것이다. 데이터셋을 9:1 로 나누어 모델을 트레이인 할 때

사용하고 나머지 1 을 validation dataset 으로 평가한다. 평가 결과는 이미지 당 [예측한 값과 / 정답 값] 으로 구성되어 있다. 이후에 validation dataset 이 아닌 코미디언들의 이미지 데이터셋을(test set) 이용하여 모델을 테스트 한다.

참고로 case 마다 있는 식은 관련 연구[1]에서 소개한 기본 코드를 기반으로 레이어의 추가, 비율 변경 등의 하이퍼 파라미터를 변경한 것이다. 특징적인 변경 식만 포함 하였으므로 자세한 코드는 해당 주소[6]에서 확인할 수 있다.

(Case1)

Conv2d * 3

Conv2d * 3 을 적용한 결과이다. Train loss 가 줄어들에 따라 validation loss도 줄어들어야 한다. 하지만 아래의 validation loss는 epoch 초반에 늘어나는 경향이 있고, 값이 너무 낮아졌다가 다시 높아지는 과정을 반복한다. Val loss = 0.2446

{0.83226494303074094, 0.5250976503738249, 0.5102616272791467, 0.4982299573734553, 0.48535468396514353, 0.49013958116974493, 0.472221187750499, 0.458716597111288, 0.47187431590725676, 0.4498923626962608} {0.26434122598609924, 0.3063856065273285, 0.341034587097168, 0.3689832385354706, 0.24459929764270782, 0.446544359588623, 0.31249502301216125, 0.452408280849457, 0.277646690670709, 0.3708922564983368}

Fig. 3-1. Training and Validation loss about case 1 ([training loss] [validation loss])

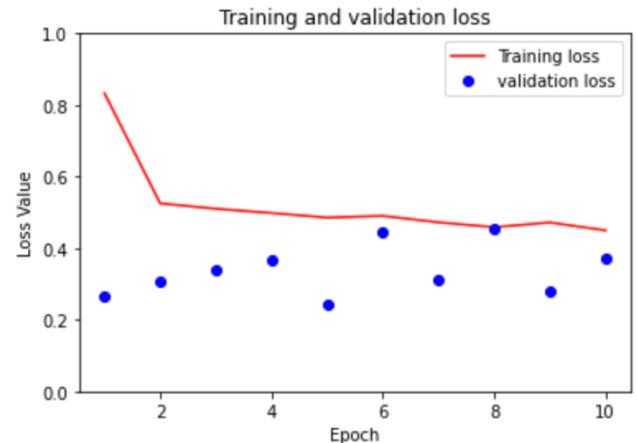


Fig. 3-2. The image of Training and Validation loss about case 1



Fig. 3-3. The images of Evaluate model about case 1



Fig. 3-4. The images of Test model about case 1

Ranking: 길 – 형돈 – 하하 – 홍철 – 준하 – 재석 – 명수

(Case2)

Conv2d * 6

Case1 보다 훨씬 validation loss의 흐름이 나아졌다. train loss가 줄어들에 따라 validation loss도 줄어드는 경향을 볼 수 있다.

val loss: 0.2262

[0.67921038080581639, 0.553091534713302, 0.4987062755257192, 0.496261338077139344, 0.48067975535537255, 0.4669242979540969, 0.450395864012994, 0.45038925210634867, 0.4418797562579916, 0.442236494969936] [0.7362799644470215, 0.360143780708313, 0.3350338933852051, 0.4005367457866687, 0.306473970413208, 0.3271717131137848, 0.3574359118938446, 0.4350733458995819, 0.22616355121135712, 0.3187032639980316]

Fig. 4-1. Training and Validation loss about case 2 ([training loss] [validation loss])

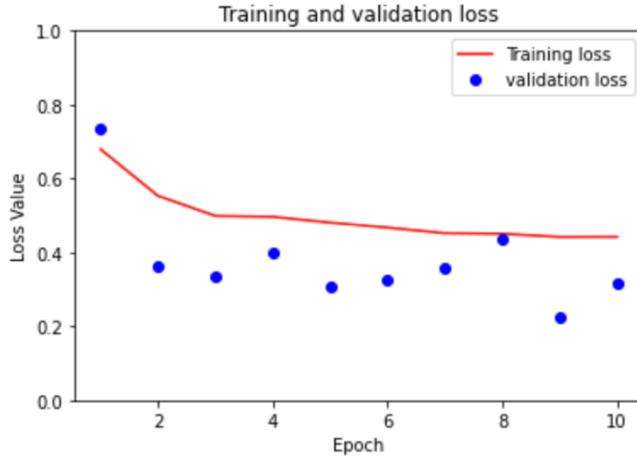


Fig. 4-2. The image of Training and Validation loss about case 2

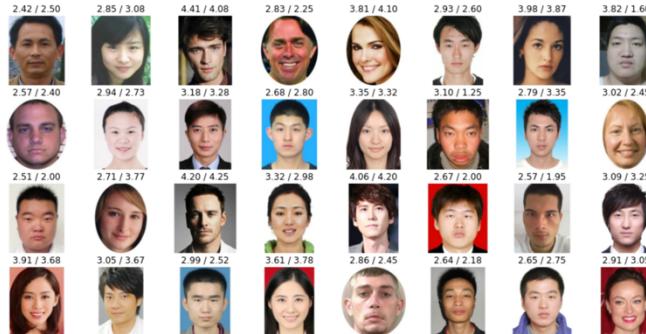


Fig. 4-3. The images of Evaluate model about case 2



Fig. 4-4. The images of Test model about case 3

Ranking: 길 – 홍철 – 하하 – 재석 – 형돈 – 길 – 준하 – 명수

(Case3)

imageDataGenerator change based on layer in case 2
rotation_range=50
width_shift_range=0.30
height_shift_range=0.30
shear_range=0.2
zoom_range=0.4
Conv2d * 6

이번에는 case2 레이어를 기반으로 각종 ImageDataGenerator 값을 변경시켜보았다. 뒤죽박죽 알 수 없는 흐름이다. case1 과 2 보다 더 좋지 않아 보인다.

val loss: 0.4524

[0.6742059834317834, 0.5628591322176384, 0.5484012696357689, 0.5113047202548595, 0.5025192235214542, 0.5101945538593061, 0.5105856264721244, 0.5095534021926649, 0.4877849561758716, 0.4673718965173972] [0.593719780445998, 0.6198229789733887, 0.627571921395984, 0.7144801616668701, 0.46123480796813965, 0.8222835659980774, 0.529579043383667, 0.4992748200893402, 0.45243561267852783, 0.52929180606051941]

Fig. 5-1. Training and Validation loss about case 3 ([training loss] [validation loss])

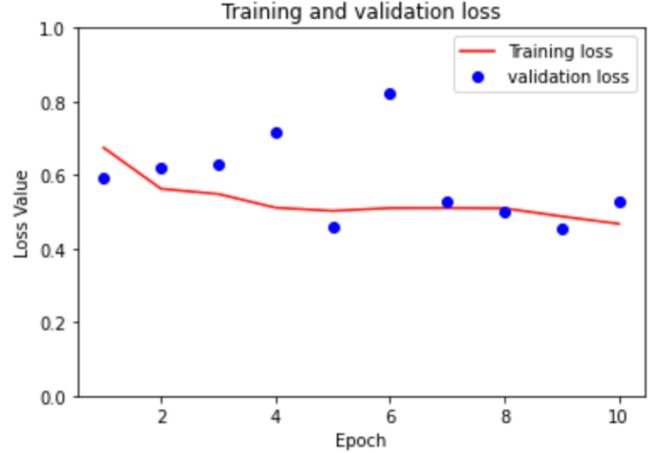


Fig. 5-2. The image of Training and Validation loss about case 3



Fig. 5-3. The images of Evaluate model about case 3



Fig. 5-4. The images of Test model about case 3

Ranking: 홍철 – 하하 – 재석 – 형돈 – 길 – 준하 – 명수

(Case4)

Conv2d * 6 + dropout * 3

Case3 레이어를 기반으로 이번에는 3,4,5 Conv2d layer의 뒤에 dropout을 적용시켜보았다. train loss가 줄어들에 따라 validation loss도 줄어드는 경향을 볼 수 있다. dropout의 기법을 통해 case3 보다 훨씬 나아졌음을 확인할 수 있다.

val loss: 0.4612

[0.6595991791378368, 0.553007978816031, 0.5536498987912894, 0.5225617820301131, 0.5085777094629076, 0.513763377618308, 0.5105055599864, 0.498859525279239, 0.48893399772096853, 0.4420923103811, 0.43603038746452, 1.007323262756839, 0.7157788692155457, 0.75055742284394, 0.4277645826339722, 0.6370517611503601, 0.5512711405754089, 0.3657167434692383, 0.46117743849754333, 0.5253857970237732]

Fig. 6-1. Training and Validation loss about case 4 ([training loss] [validation loss])

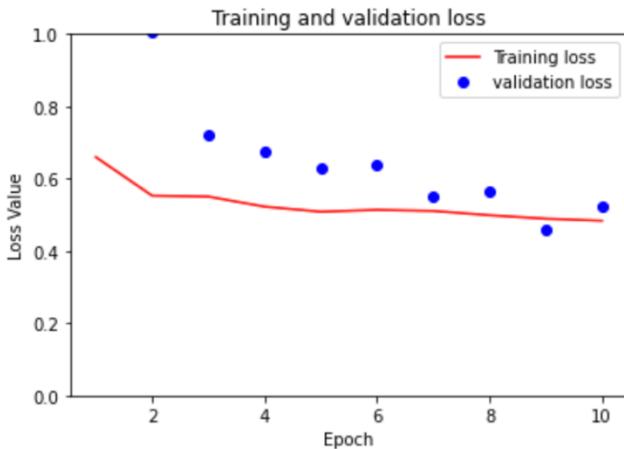


Fig. 6-2. The image of Training and Validation loss about case 4



Fig. 6-3. The images of Evaluate model about case 4



Fig. 6-4. The images of Test model about case 4

Ranking: 형돈 – 홍철 – 하하 & 재석 – 준하 – 길 – 명수

(Case5)

Optimizer change based on layer in case 4 Adam -> RMSProp

케이스 4의 레이어를 기반으로 옵티마이저 변경해봤다.
전체적인 validation loss 가 작아졌다.
val loss: 0.3724

[0.980824101465842, 0.621724367086479, 0.4623766515798494, 0.54668600002031037, 0.5429342615965641, 0.5353934058516913, 0.5292818186013, 0.4315622746944225, 0.49051924403644714, 0.4204985797405243, 0.49227774143218994, 0.498655050995, 11.3135472536097036, 0.398514171828492, 0.4983891039875, 0.39515411853790283, 0.38093921542167664]

Fig. 7-1. Training and Validation loss about case 5 ([training loss] [validation loss])

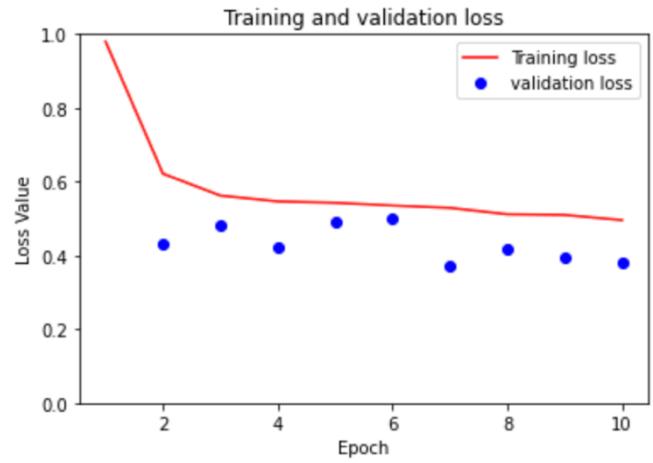


Fig. 7-2. The image of Training and Validation loss about case 5



Fig. 7-3. The images of Evaluate model about case 5



Fig. 7-4. The images of Test model about case 5

Ranking: 재석 – 형돈 – 홍철 – 하하 – 길 – 준하 -명수

(Case6)

*Conv2d * 6 + dropout * 6*

Epoch 초기에는 training loss 를 따라 줄어들었지만 그 이후에 증가하는 경향이 있다. 특히 Epoch 의 8-10 사이에서 갑자기 오버피팅이 일어나고 있다.
val loss: 0.5049

[0.6977765210951218, 0.5732310394084815, 0.5356153009756647, 0.5288574676080183, 0.5270895488334425, 0.5117029000893988, 0.505875647333427, 0.5009962457358235, 0.503314518361771, 0.505425832765271] [1.4283408644226074, 0.54199393965620178, 0.504934965641785, 0.596617978096008, 0.5608757734298706, 0.6294409036636353, 0.5875834822654724, 0.7309942245483398, 0.800582408905293, 0.599154531955719]

Fig. 8-1. Training and Validation loss about case 6 ([training loss] [validation loss])

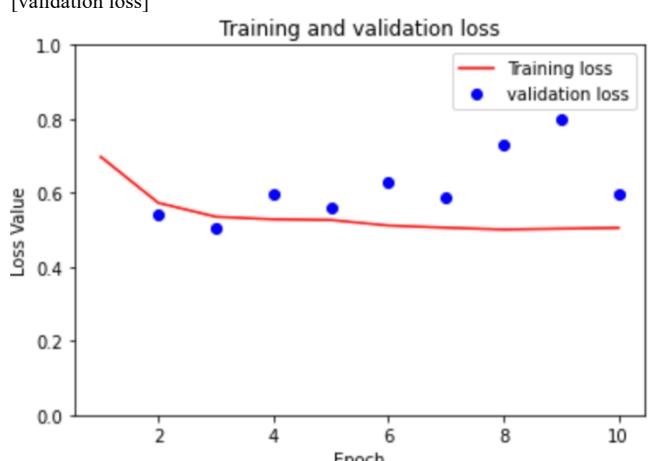


Fig. 8-2. The image of Training and Validation loss about case 6

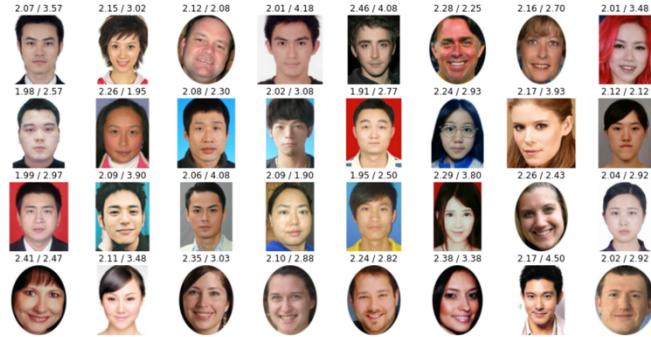


Fig. 8-3. The images of Evaluate model about case 6



Fig. 8-4. The images of Test model about case 6

Ranking: 홍철 - 재석&형돈 - 하하&준하 - 길

(Case7)

**Conv2d * 10 + dropout * 6
Adam
Dense 256 -> 512**

이번에는 레이어를 10개로 늘렸다. 그리고 dropout 을 추가했다. train loss 가 줄어들에 따라 validation loss 도 줄어드는 경향을 볼 수 있다.

val loss: 0.4124

[0.71256456724321, 0.5912385823918178, 0.5498162525350397, 0.5236366970852168, 0.5077337520531934, 0.506712292059503
4, 0.5071640799382721, 0.5060199273716319, 0.4885826358891497, 0.4907239822859714]
[0.7754789590835571, 1.3041049242019653, 0.67093834400177, 0.5549302101135254, 0.42117762565612793, 0.44203814864158
63, 0.4124360845336914, 0.4985553324225647, 0.525626597886658, 0.4773184359073639]

Fig. 9-1. Training and Validation loss about case 7 ([training loss]
[validation loss])

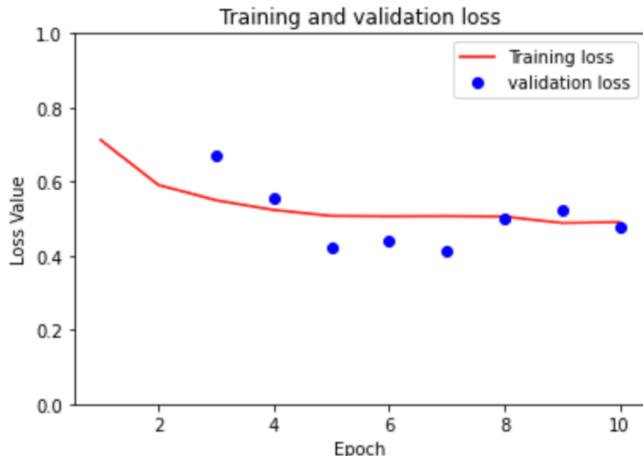


Fig. 9-2. The image of Training and Validation loss about case 7

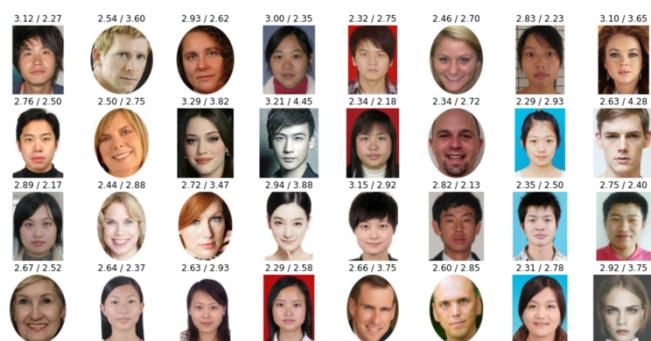


Fig. 9-3. The images of Evaluate model about case 7



Fig. 9-4. The images of Test model about case 7

Ranking: 재석-형돈-홍철-하하-준하-명수-길

(Case8)

change based on layer in case 7

Dropout 0.2 -> 0.3

Epoch 10 -> 20

케이스 7의 레이어를 기반으로 dropout 의 비율을 올리고, epoch 의 갯수도 10에서 20으로 변경했다.

val loss: 0.2114

[0.702565326150143, 0.5789105384277575, 0.5330080891980066, 0.526651415222823, 0.52141344787077
9, 0.5154912979193408, 0.5162038553573386, 0.488875340646026, 0.4974924222929244, 0.48084193783577045, 0.4766195974
4877021484409, 0.4774924222929244, 0.460866895555697778, 0.4529307646298534, 0.4469486928949527, 0.458
41198723252596, 0.4478597826548297, 0.4373469427108765, 1.4345165491104126, 1.015865468979882,
[2.2572021484375, 1.801342487335295, 1.4395469427108765, 1.4345165491104126, 1.015865468979882,
0.664485592842102, 0.536172091969097, 0.596894443035125, 0.403106395816803, 0.5155561566353284
4, 0.5361345410346985, 0.30134761333465576, 0.3097754716873169, 0.4834709465503692, 0.4114149808883667, 0.2133114784
959784, 0.42634761333465576, 0.21140675246715546]

Fig. 10-1. Training and Validation loss about case 8 ([training loss]
[validation loss])

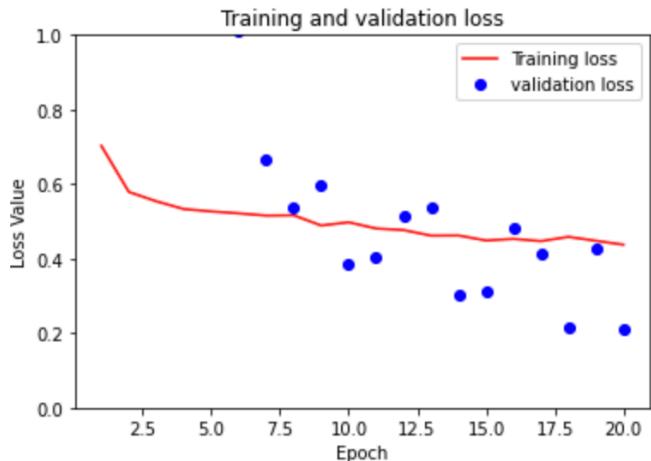


Fig. 10-2. The image of Training and Validation loss about case 8



Fig. 10-3. The images of Evaluate model about case 8



Fig. 10-4. The images of Test model about case 8

Ranking: 길 - 홍철 - 하하&준하 - 재석 - 형돈&명수

D. Result

a) *Positioning Figures and Tables:* 실제 무한도전 프로그램에서의 최종 투표 결과이다. 현장투표, 온라인, 전문의, 해외 각 투표 결과를 동일한 비율(각 25%)로 반영하여 합산한 최종 결과이다.

TABLE I. RANKING COMEDIANS FACE

최종순위	멤버	현장투표	온라인	전문의	해외	점유율
1	홍철	2	1	1	1	40.1%
2	재석	1	2	3	3	20.6%
3	하하	3	3	2	2	19.0%
4	길	7	6	4	4	5.9%
5	형돈	5	4	7	5	5.5%
6	준하	6	5	5	7	4.9%
7	명수	4	7	5	6	4.7%

a. 무한도전 최종 결과[5]

그 중에서도 특히 val_loss 가 낮았던 Case 8 의 model에 적용한 결과와 비교해 보려고 한다.



Ranking: 길 - 홍철 - 하하 & 준하 - 재석 - 형돈 & 명수

홍철이 상위권에 위치한 것과, 명수가 최하위 점수를 기록했다는 점이 비슷하지만 나머지 부분에서 공통점을 찾기 힘들다. 추가적으로 case 8 개중 2 개 이상에서 최하위 포지션 중 공통적으로 명수의 얼굴이 나온 것이 흥미로웠다.

V. CONCLUSION

본 연구에서는 Supervised Learning 인 딥러닝 모델을 사용했다. Supervised Learning 는 데이터에 대한 정답이 있고, 정답에 맞게 모델을 훈련 시키는 방법이다. 따라서 SCUT-FBP5500 Dataset 를 이용하여 해당 얼굴 데이터에 beauty_socre 을 레이블링 한 값으로 훈련을 시켜 모델을 만들었다. 최종적으로 딥러닝을 이용하여 얼굴 평가해주는 기계를 만든 것이다.

Case 경우 별로 train loss 와 validation loss 를 비교해봤다. Conv layer 를 추가해보고, imageDataGenerator 값도 바꿔보았다. 마지막 케이스에서는 epoch 의 숫자도 조절해봤다.

case8 모델은 val_loss 가 0.2114 로 가장 낮았다. 해당 결과를 최종적으로 실제 무한도전 프로그램의 투표 결과와 비교했다. 무한도전 프로그램의 최종 랭킹은 홍철-재석-하하-길-형돈-준하-명수 순이었다. 그리고 모든 모델의 순위는 길 - 홍철 - 하하 & 준하 - 재석 - 형돈 & 명수였다. 뒤죽박죽 순서가 맞지 않았다. 결과가 다른 것은 어떻게 보면 당연한 것이다. 모델을 학습시키기 위해 사용된 데이터 자체가 주관적이기 때문이다.

추가적으로 transfer learning 을 적용해보지 못한 아쉬움이 있다. 기존에 존재하는 모델로부터 새로운 모델을 학습시켜 조금이라도 더 좋은 결과가 나오지 않았을까라는 생각이 들었다.

이 연구를 더 발전 시켜 꼭 코미디언들의 얼굴이 아니더라도 여러 마케팅 전략에 활용할 수 있을 것 같다고 생각했다. ‘미국에서 선호하는 얼굴’, ‘중국에서 선호하는 얼굴’, ‘한국에서 선호하는 얼굴’ 등 의 실험을 진행하여 해당 문화권을 타겟으로 한 문화 컨텐츠 사업으로 이용하면 좋을 것 같다.

REFERENCES

- [1] https://github.com/WonJunPark/beauty_score
- [2] LIANG, Lingyu, et al. Scut-fbp5500: A diverse benchmark dataset for multi-paradigm facial beauty prediction. In: 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018.
- [3] <http://www.imbc.com/broad/tv/ent/challenge/event/minam/index.html>
- [4] <https://github.com/HCIILAB/SCUT-FBP5500-Database-Release>
- [5] <https://namu.wiki/w/%EB%AC%B4%ED%95%9C%EB%8F%84%E%C%A0%84%20%EB%AF%B8%EB%82%A8%EC%9D%B4%EC%8B%9C%EB%84%A4%EC%9A%94>
- [6] <https://github.com/leyuri/Ranking-Comedians-Face>