Final Report

2020095178 최윤선
2021029443 전예은

[Abstract]

These days many people use blind date apps to find their ideal person. Other person's appearance is one of the important criteria when determining to meet a new person. In the same time, people are afraid to show their own appearance. So we thought about deep learning strategies to explain the people's appearance without seeing the photo. In the project we used CelebA datasets, and preprocessed the data manually. We have thought about 2 deep learning methods: Object Detection and Image Captioning. For object detection, we used MTCNN and Faster R-CNN, and for image captioning, we used Resnet152-LSTM and Transformer. By object detection, people can choose desired skin color and desired ratio of a nose and lips of their ideal person. By image captioning, people can see the description of the other's appearance without seeing the image.

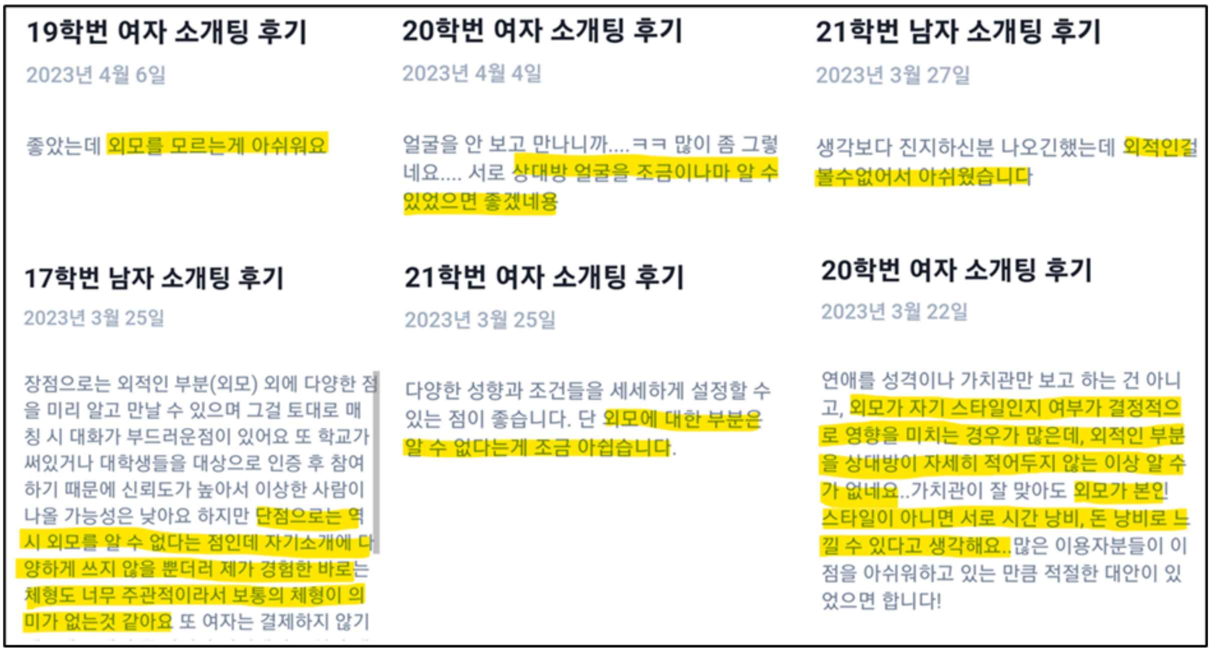
1.Introduction

1.1. Project Goal

The big goal of our project is to **"Get to know the image in a situation where we can't see the image."**
These days, many people use blind date apps to find their ideal person. There are several requirements they want from future partners such as academic background, economic ability, residence, and appearance. Among these many factors, people are afraid to show their appearance by photo because someone else using the same dating app might recognize their face. At the same time, people are curious about the appearance of the person they will meet on the first day of blind date. So we are going to design some algorithms that match the future partner who has the appearance the users want and also objectively explain the appearance of the future partners.

Why appearance?
It is true that we are building algorithms that focus on people's appearance, but we are not saying that appearance is the most important factor when finding a future partner. We think that appearance is just one factor that people consider when looking for a future partner among many other factors such as academic background, economic ability, residence etc. The reason that we are focusing on appearance is because other factors are easy to express by writing, but it is not easy to express appearance by writing or quantitative methods. In addition, it is difficult to describe one's face objectively on one's own (for example, skin color and eye size). Since many people have the perception that artificial intelligence is more objective than humans, if AI objectively describes people's appearance rather than people explaining their appearance by themselves, users will put more trust into the description of appearance.

Is appearance necessarily important?
Even if it is not a blind date, it is undeniable that many people are unconsciously evaluating others' appearance. Especially in the case of blind dates, appearance can be an important factor in developing feelings about future partners.
“Yeonpick” was one of the light blind date apps that users cannot know much about others’ appearance. In the app, people could only explain their appearance by selecting choice of some simple characteristics such as double eyelids or not. There were many reviews of the app saying that it’s too bad that they can’t know the partner’s appearance until they meet in person[Figure1].



19학번 여자 소개팅 후기
2023년 4월 6일
좋았는데 외모를 모르는게 아쉬워요

20학번 여자 소개팅 후기
2023년 4월 4일
얼굴을 안 보고 만나니까....ㅋㅋ 많이 좀 그렇네요.... 서로 상대방 얼굴을 조금이나마 알 수 있었으면 좋겠네용

21학번 남자 소개팅 후기
2023년 3월 27일
생각보다 진지하신분 나오긴했는데 외적인걸 볼수없어서 아쉬웠습니다

17학번 남자 소개팅 후기
2023년 3월 25일
장점으로는 외적인 부분(외모) 외에 다양한 점을 미리 알고 만날 수 있으며 그걸 토대로 매칭 시 대화가 부드러운점이 있어요 또 학교가 써있거나 대학생들을 대상으로 인증 후 참여하기 때문에 신뢰도가 높아서 이상한 사람이 나올 가능성은 낮아요 하지만 단점으로는 역시 외모를 알 수 없다는 점인데 자기소개에 다양하게 쓰지 않을 뿐더러 제가 경험한 바로는 체형도 너무 주관적이라서 보통의 체형이 의미가 없는것 같아요 또 여자는 결제하지 않기

21학번 여자 소개팅 후기
2023년 3월 25일
다양한 성향과 조건들을 세세하게 설정할 수 있는 점이 좋습니다. 단 외모에 대한 부분은 알 수 없다는게 조금 아쉽습니다.

20학번 여자 소개팅 후기
2023년 3월 22일
연애를 성격이나 가치관만 보고 하는 건 아니고, 외모가 자기 스타일인지 여부가 결정적으로 영향을 미치는 경우가 많은데, 외적인 부분을 상대방이 자세히 적어두지 않는 이상 알 수가 없네요..가치관이 잘 맞아도 외모가 본인 스타일이 아니면 서로 시간 낭비, 돈 낭비로 느낄 수 있다고 생각해요..많은 이용자분들이 이 점을 아쉬워하고 있는 만큼 적절한 대안이 있었으면 합니다!

Figure1. Reviews of blind date app “Yeonpick”

By these reviews, we can see that there are many people who value appearance before going on a blind date.
In a society where one is reluctant to show one’s appearance by photo, we would like to suggest a way to grasp others’ appearance without seeing the photo. We expect that users will have more trust in the description of appearance by AI’s objective explanation rather than people’s subjective explanation.

1.2. Project Design Sketch

Our goal is to design the algorithm that matches the future partner who has the appearance the user wants and besides objectively explains the appearance of the future partner. We hope that this algorithm could be included in the future blind date apps. The overall process of the algorithm we planned to make is like this[Figure2].
(1) User uploads photo.
The user uploads a photo that shows the user’s appearance. We will let users know that the photo will not be released anywhere else. The other users cannot see the photo, and we have access to the photo but we will not see the users’ photo also for personal information protection.
(2) Classify the user’s photo using object detection.
Let’s think that there are facial attributes that the user can choose for a future partner. By using object detection, we can extract the facial attributes. By getting the facial attributes, we classify the photo based on some fixed standard.

(3) Label the photo.
Since we can classify these photos by object detection, we can label them with those facial attributes. By this process, we will have a huge number of users' photos with labels.

(4) Match the photo that ideal type is each other.
Now let's go back to the user's point of view. Users will choose the desired facial attributes they prefer for future partners. We select all the photos that have desired facial attributes that the user wants, and between those photos we would select photos whose ideal type is the user. Between those photos, we will select one random photo and match with the user.

(5) Get more information about the appearance by word explanation.
The user will also be curious about the additional appearance. We will use image captioning to explain the photo. Let's say that the result of the image caption of the man's photo will be like "A man with a big nose and curly dark black hair is wearing a suit." Then we will show the image caption result to the woman, not the photo. Women can know more about the external appearance of the man by the image caption sentence and be curious about the man.

(6) Decide to meet each other.
Same thing happens on both man and woman. If they both choose to meet each other, they meet. If one of them chooses not to meet, they don't meet and we introduce a new person by repeating the same process above.

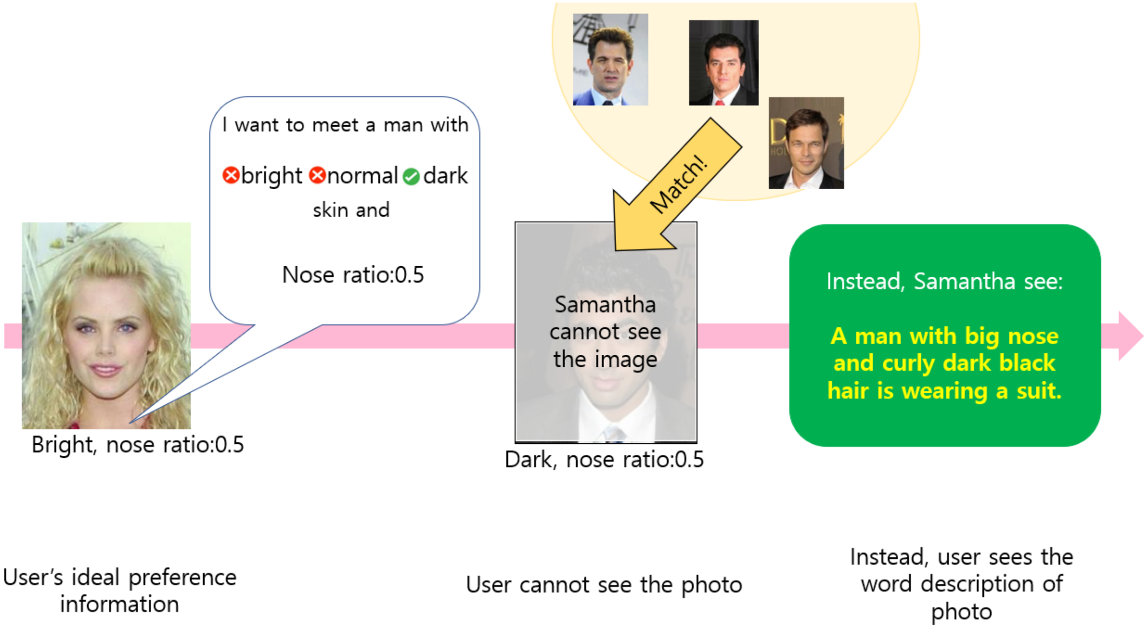

I want to meet a man with
☒bright ☒normal ✓dark
skin and
Nose ratio:0.5
Bright, nose ratio:0.5
Match!
Samantha cannot see the image
Dark, nose ratio:0.5
Instead, Samantha see:
A man with big nose and curly dark black hair is wearing a suit.
User's ideal preference information
User cannot see the photo
Instead, user sees the word description of photo

Figure2. Overall process

2. Previous Works

Computer Vision is one of the most actively researched areas. In particular, related to Image detection and Image captioning, there is a significant amount of applications and research that improves the performance of the models.

We'll be dealing with facial images to create a demo of the blind date app, so we've looked up some related works for this project and have written reviews in the Notion.

2.1. Image Detection

There are classic models such as R-CNN, Fast R-CNN, Faster R-CNN, and YOLO for image detection. Also, detection models using Attention techniques have recently performed well. Facial image detection studies using the image detection methods are being conducted, and related models include MTCNN or RetinaFace, etc. With these image detection models, we can do several tasks such as detecting the face itself from on image and extracting facial landmarks such as eyes, nose, and mouth from the face.
In this project, we implemented the Faster R-CNN[1][Figure3] and MTCNN[2][Figure4] models that can extract facial landmarks among existing studies because we need facial features. Below are the model architectures proposed in the previous works.

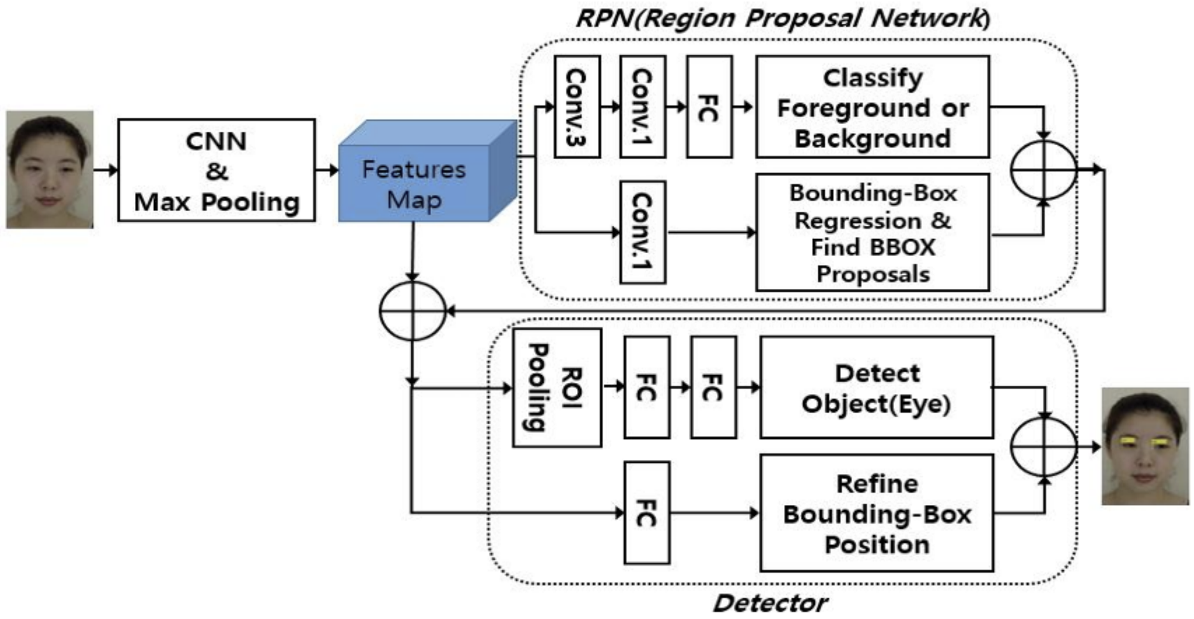

RPN(Region Proposal Network)
CNN & Max Pooling
Features Map
Conv.3
Conv.1
FC
Classify Foreground or Background
Conv.1
Bounding-Box Regression & Find BBOX Proposals
ROI Pooling
FC
FC
Detect Object(Eye)
FC
Refine Bounding-Box Position
Detector

Figure3. Faster R-CNN model architecture

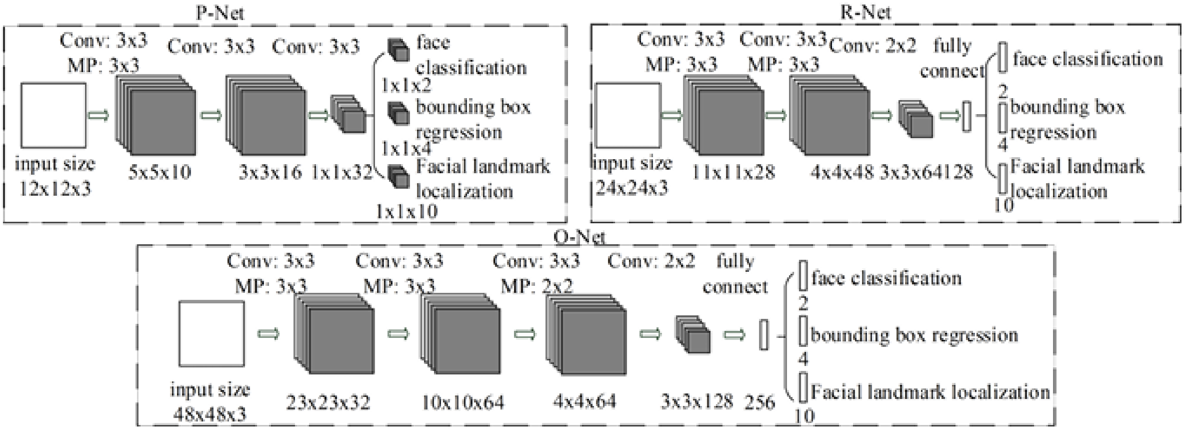


Figure4. MTCNN model architecture

2.2. Image Captioning

Basically, image captioning models consist of a CNN encoder architecture that extracts the feature of the image and an RNN decoder architecture that generates the captioning. Current research improves CNN encoder and RNN decoder to perform better. CNN part has been modified to VGG, GooLeNet, ResNet, etc., and RNN part has been modified to LSTM, etc. to improve performance. In addition, some image captioning models using the Transformer with attention modules are also studied. In this project, we implement and compare basic ResNet-LSTM based model[3][Figure5] with Transformer-based model[4][Figure6], which is an application model. Below are the model architectures proposed in the previous works.

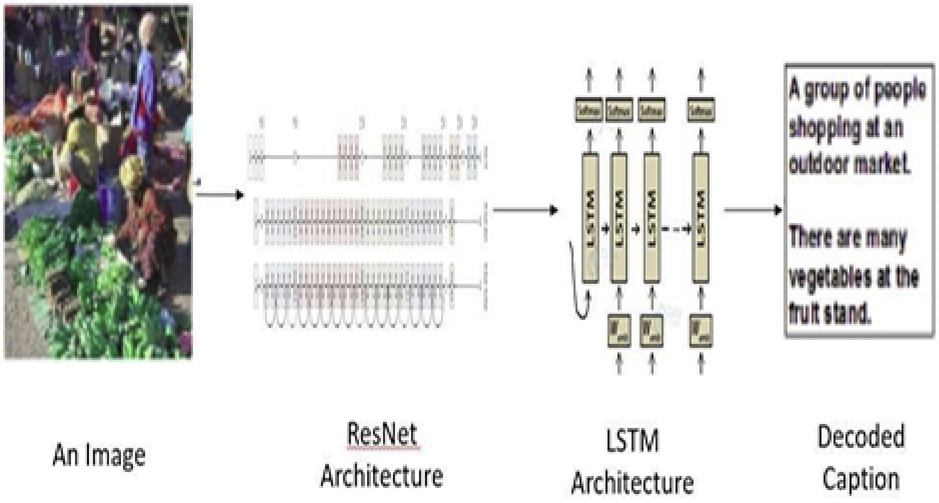

A group of people shopping at an outdoor market.
There are many vegetables at the fruit stand.
LSTM
An Image
ResNet Architecture
LSTM Architecture
Decoded Caption

Figure5. ResNet-LSTM model architecture

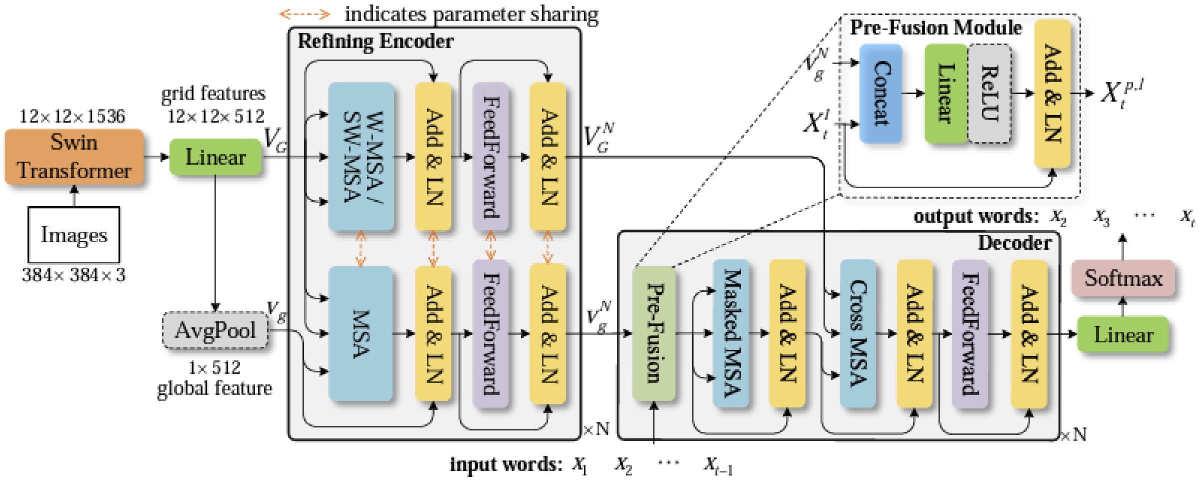


Figure6. Transformer-based model architecture

3. Proposed Method

3.1. Data

We used CelebA dataset(https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html). CelebA dataset contains 202,599 face images of the size 178×218 from 10,177 celebrities, each annotated with 40 binary labels indicating facial attributes like hair color, gender and age. We chose this dataset because people's face sizes were adjusted to the similar scale, and there were binary labels about facial attributes that could help us in data preprocessing.

3.2. Data Preprocessing

We determined that this dataset was too large to train the model, so we decided to select 10000 woman images and 10000 man images. We decided to select the good quality

images by looking at the image one by one and choose the images that will well fit in the model. We had some standard in selecting the images:
1) The image should not be black and while and the image should not be blurry.
2) The person should be looking at the front, not the side.
3) The person should not have heavy makeup, and the hair should not hide their facial attributes such as eyes and lips. Also, if the accessories hide the facial attributes (sunglasses hiding eyes, hats hiding eyes), we should not select the image.
4) Choose the image of the person that is expressionless as possible. If the person is smiling or frowning, because of the facial expression, it might lower the accuracy of the model.
5) We should not choose the person who is too young or too old. People within these ages have high possibility of not using the blid date apps.

3.3. Method and Algorithm

We had 2 explicit goals on this project.
1) Goal 1: Digitialize some facial attributes.
We thought that if we digitalize some facial attributes, we could classify the images and help people find the ideal person automatically by using the algorithm.
In order to extract facial attributes from the image, we thought that we need to do object detection on the image first. We decided to do object detection with different models and individually think about further analysis of digitalization after doing the object detection. Yoonseon performed object detection with MTCNN and Yeeun performed object detection with Faster R-CNN.
2) Goal 2: Explain the person image by words.
We thought that explaining the person image by words could solve the image privacy problem. People don’t want to show their personal image to others, and if the image can be explained by words, this problem will be solved.
Goal2 could be done by image captioning. To perform image captioning, the corresponding caption txt files are required. However, the CelebA dataset does not include the caption data. We judged that it was impossible to manually create caption data for a total of nearly 20,000 image data. So we pre-trained the model using the Flickr8k dataset including caption data with the training dataset of the image captioning models, and then test with the CelebA dataset only.
We decided to do image captioning with different models and compare the results. Yoonseon performed image captioning with ResNet152-LSTM and Yeeun performed image captioning with transformer.

4. Analysis and Results

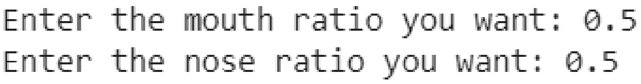
4.1. Yoonseon

4.1.1. Object detection (MTCNN)

First, for facial landmark detection, I implemented MTCNN by referring to github code[5]. However, unknown errors about the code continued to occur, and I determined that matching the input data set form of the CelebA dataset for the implemented model would be extremely complicated and time-consuming. Therefore, the code I implemented based on the paper could not be used, and I conducted facial detection by installing the MTCNN package. The code I implemented was attached to the [AI_project] link of notion.
Using the CelebA images, I find the bounding boxes of the face, the coordinate points of the eyes, nose, and mouth through the MTCNN detector. Using these coordinate points, I calculate the face size, and the ratio of the nose length and the lip length to the face size. (The specific details are like this: First, I measure the size of the face, the length of the nose,

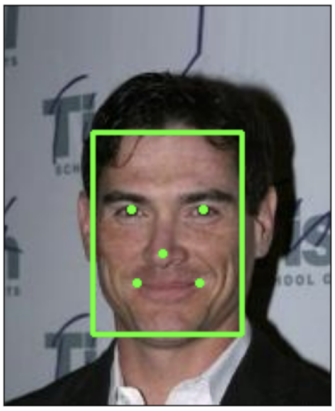
and the mouth with location information. As for the length of the nose, I compute the distance between the nose point and the straight line which is connected by the points of the two eyes. The length of the nose and the lips are divided by the height and width of the face, respectively, and the ratio is calculated.)
Based on these calculated values, when the user enters the ratio of the length of the nose and lip of the blind date that he desires, the image satisfying entered values is randomly extracted. However, during detection, several bounding boxes may be displayed because there are several faces in one image. These images are deleted because there are several facial landmarks.

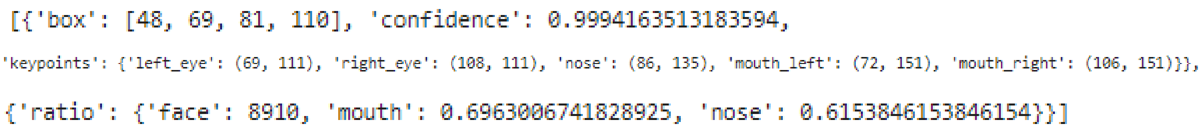
The facial image examples derived using the MTCNN detector and the coordinate values and landmark ratios are as follows[Figure7][Figure8].



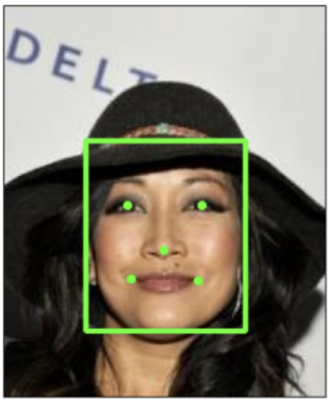
```
Enter the mouth ratio you want: 0.5
Enter the nose ratio you want: 0.5
```

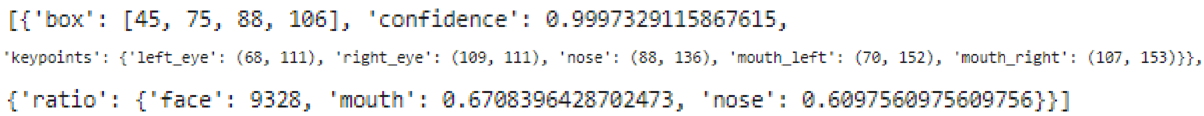
Figure7. The user can enter the ratios he wants.





```
[{'box': [48, 69, 81, 110], 'confidence': 0.9994163513183594,
'keypoints': {'left_eye': (69, 111), 'right_eye': (108, 111), 'nose': (86, 135), 'mouth_left': (72, 151), 'mouth_right': (106, 151)}},
{'ratio': {'face': 8910, 'mouth': 0.6963006741828925, 'nose': 0.6153846153846154}}]
```





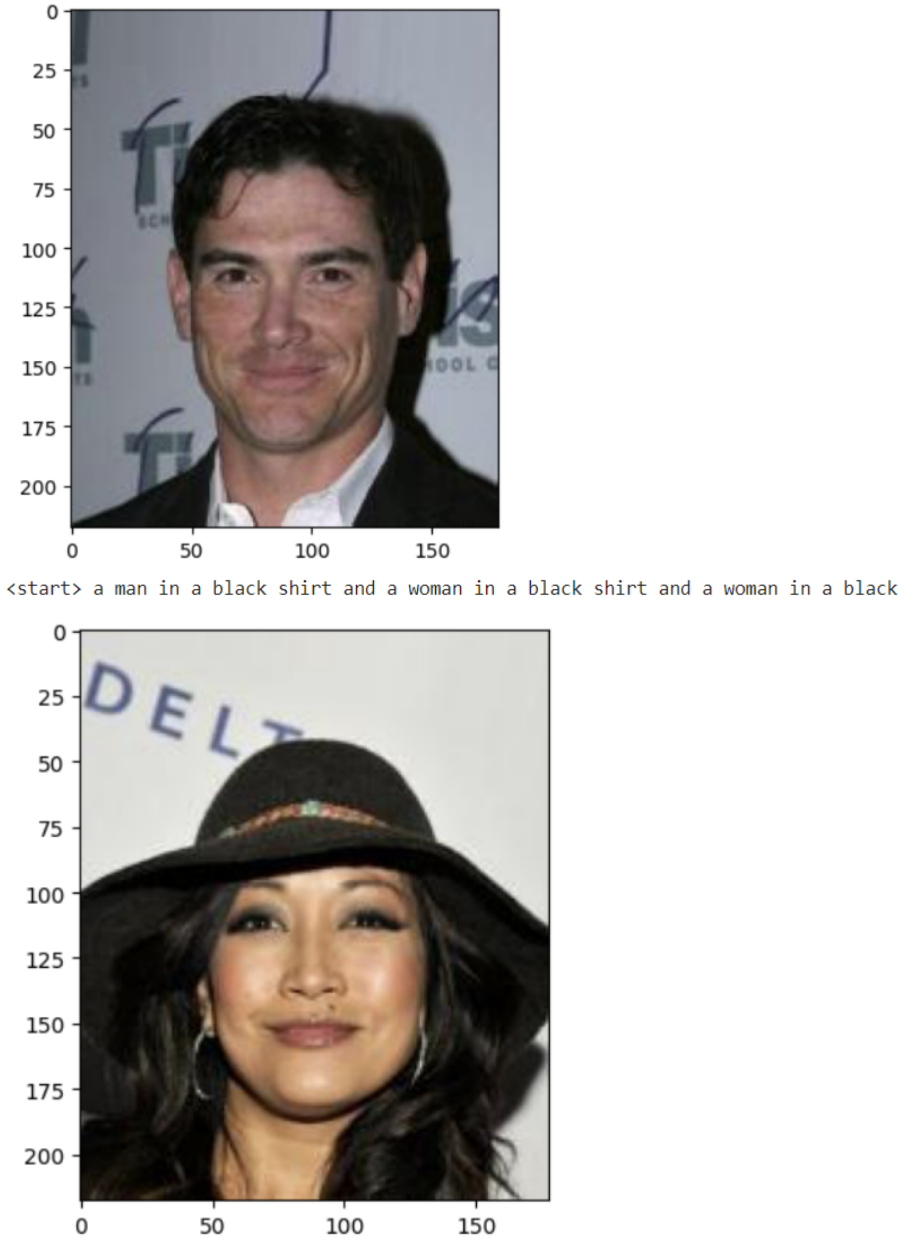
```
[{'box': [45, 75, 88, 106], 'confidence': 0.9997329115867615,
'keypoints': {'left_eye': (68, 111), 'right_eye': (109, 111), 'nose': (88, 136), 'mouth_left': (70, 152), 'mouth_right': (107, 153)}},
{'ratio': {'face': 9328, 'mouth': 0.6708396428702473, 'nose': 0.6097560975609756}}]
```

Figure8. The detected man and woman images and information

4.1.2. Image Captioning (ResNet152-LSTM)

Next, I implement an image captioning model to describe the features of the extracted image. The encoder of the model uses ResNet152, which showed good performance with deep layers in image feature extraction, and the decoder uses LSTM, which will solve the long term dependency problem, to improve the capturing performance. I use Flickr8k dataset to pre-train the model. The loss function is cross entropy loss, the learning rate is set to 0.001 and the epoch is set to 5. We also use validation dataset for performance evaluation. Finally, a caption is generated using the pre-trained ResNet152-LSTM model for images extracted through detection and ratio calculation.

The results of captioning the above two images are as follows[Figure9].



```
<start> a man in a black shirt and a woman in a black shirt and a woman in a black
```

```
<start> a woman in a black jacket and a black hat and a white shirt . <end>
```

Figure9. The man and woman images with captions

The results of detection are extracted fairly accurately from the confidence score. However, the captioning's results are not as perfect as I expected, but they explain the characteristics of the images to some extent.

4.2. Yeeun

4.2.1. Object detection (Faster R-CNN)

For object detection, I used Faster R-CNN method. I tried to implement Faster R-CNN from scratch, but it was not easy for me. Torchvision.models.detection provides Faster R-CNN API(torchvision.models.detection.fasterrcnn_resnet50_fpn), so I used it to implement Faster R-CNN.

I used Stochastic Gradient Descent for optimizer, and set learning rate 0.005, momentum 0.9, and weight decay 0.0005. And trained for 10 epochs. I calculated the bounding box of the face based on the facial attributes, and the prediction result on test dataset was like this[Figure10].

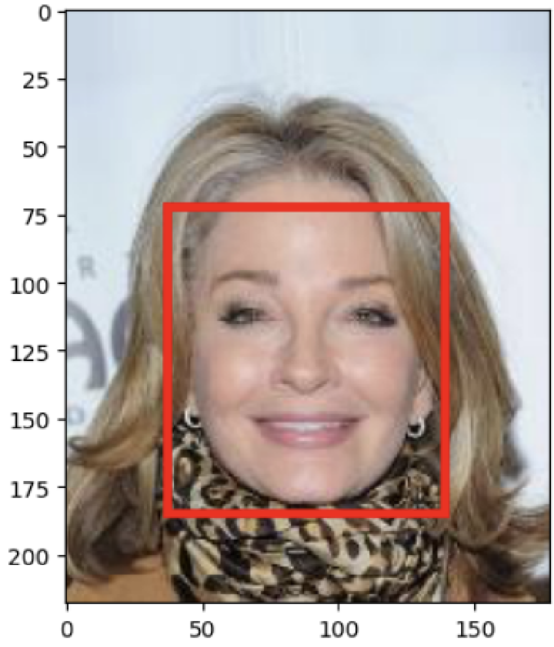
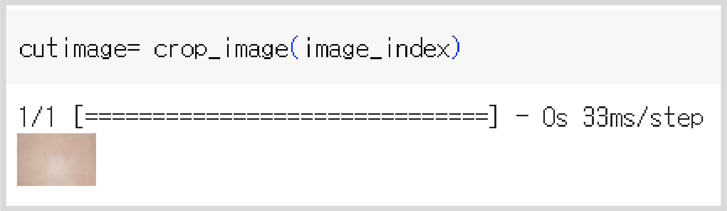


Figure10. Prediction Result

Using face detection, I thought we could extract skin color. We could crop the part of forehead and extract the skin color from it[Figure11].



```
cutimage= crop_image(image_index)
```

```
1/1 [==============================] - 0s 33ms/step
```

Figure 11. Cropped image

After changing the image to RGB scale, the luminance could be calculated like this:
Luminance= 0.2126 * R_avg + 0.7152 * G_avg + 0.0722 * B_avg

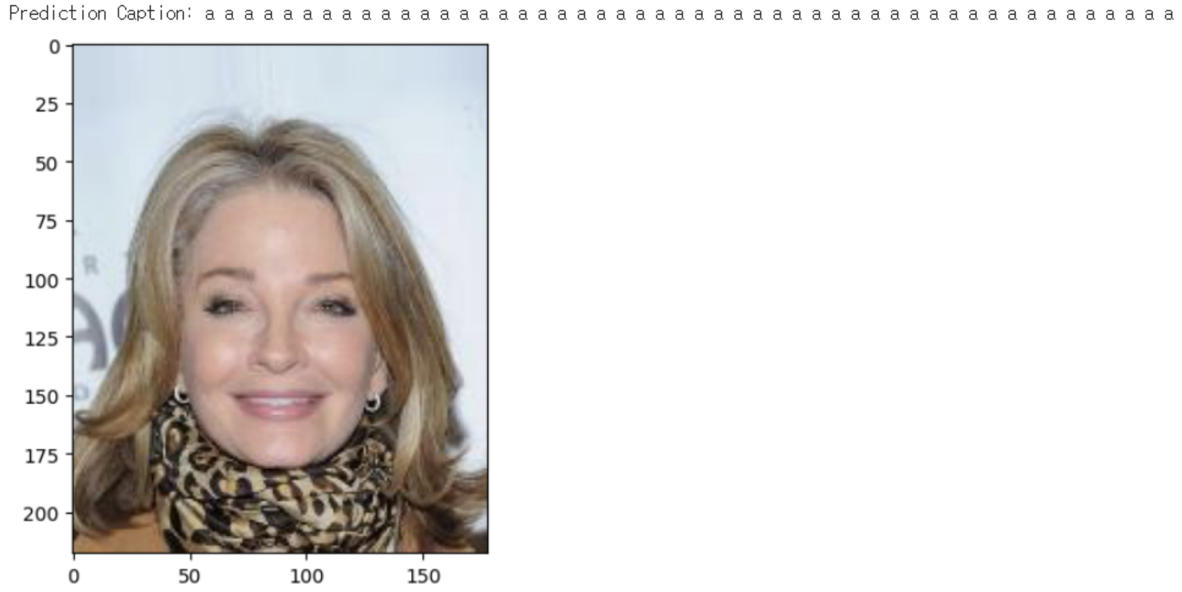
So, we can calculate the luminance of the forehead skin color, and classify based on some standard. So by this method, we can classify the people skin color into bright, normal, and dark. After classifying, we can label them bright, normal, or dark. If the user wants to meet a person who has dark skin, we can match a person who has labeled dark.

In this project, I did object detection on face, but in further projects, it would be better to do object detection on the area of forehead or cheek to extract the skin color more easily.

4.2.2. Image Captioning (Transformer)

According to [4], Transformer-based model integrates image captioning into one stage and realizes end-to-end training. In transformer-based model, adopt SwinTransformer to replace Faster R-CNN as the backbone encoder to extract grid-level features from given images.

Then, referring to Transformer, build a refining encoder and a decoder. The refining encoder refines the grid features by capturing the intra-relationship between them, and the decoder decodes the refined features into captions word by word. Furthermore, in order to increase the interaction between multi-modal (vision and language) features to enhance the modeling capability, calculate the mean pooling of grid features as the global feature, then introduce it into refining encoder to refine with grid features together, and add a pre-fusion process of refined global feature and generated words in decoder.
In this project, for transformer based image captioning, I applied EfficientNetB2 for image feature extractor (instead of SwinTransformer) and skip refining encoder. I used Flickr8k dataset to pre-train the model. I tried to train on more epochs, but I could only train on 1 epoch due to the usage limit of GPU in colab.
The result was inaccurate[Figure12].



```
Prediction Caption: a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a a
```

Figure12. Image Captioning Result by transformer
All the prediction Caption was written with ‘a’. I could not find the reason for the inaccurate result from analyzing the code, so I thought it was due to the small number of epochs.

5. Conclusion

We did object detection by MTCNN and Faster R-CNN. We were able to detect the face by both methods. In both methods, the result was accurate in several images. By object detection, we can calculate ratio of nose length and lips length, and calculate the luminance of skin color. Based on the calculated result, we can classify the photo and match people easily with their desired ideal type automatically.
We did image captioning by Resnet152-LSTM and Transformer. Our initial goal was to compare the results by different models, however the Transformer result was inaccurate. Also, the result of ResNet152-LSTM was not quite perfect. So we are just proposing that there are methods to explain the people’s appearance by models such as Resnet152-LSTM and Transformer.

Reference

[1] A Method of Eye and Lip Region Detection using Faster R-CNN Face Image
https://koreascience.kr/article/JAKO201827041051649.pdf

[2] Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks
https://arxiv.org/ftp/arxiv/papers/1604/1604.02878.pdf

[3] Image Caption Generator Using RESNET-LSTM
https://www.ijres.org/papers/Volume-9/Issue-8/Series-1/L09086471.pdf

[4] End-to-End Transformer Based Model for Image Captioning (Yiyu Wang)
https://arxiv.org/pdf/2203.15350.pdf

[5] https://github.com/timesler/facenet-pytorch

Notion Link

[AI_project] Display model implementation progress and dates
https://pale-baron-107.notion.site/8955286c42c346cab41847e49038e333?v=c59e06af6aea4192b3bd78a4f771c9c6&pvs=4

[Document] Upload code, report, paper review, ppt
https://pale-baron-107.notion.site/4999fd982ed5477d96f9e02918e523b4?v=180e9ecdf2774c0db4f91f52113244e3&pvs=4

[Meeting] Contents of the meeting
https://pale-baron-107.notion.site/8c4aa63b432c44b9a65b112810f66507?v=f136a1c573d44c00be3f138dd6724f2a&pvs=4