Artificial Intelligence Project
Final Presentation

<Topic>

Demo of Blind Data App Using Facial Detection & Image Captioning Models

2020095178 YoonSeon Choi (최윤선)
2021029443 YeEun Jeon (전예은)

Index

Part1. Introduction: Project Goal

Dating apps…

I want my future partner to be:
- 27-29 years old
- School above Hanyang University
- Etc..

Hmm.. I am afraid to show my photo, but I am curious about the appearance of the man that will show up in blind date.

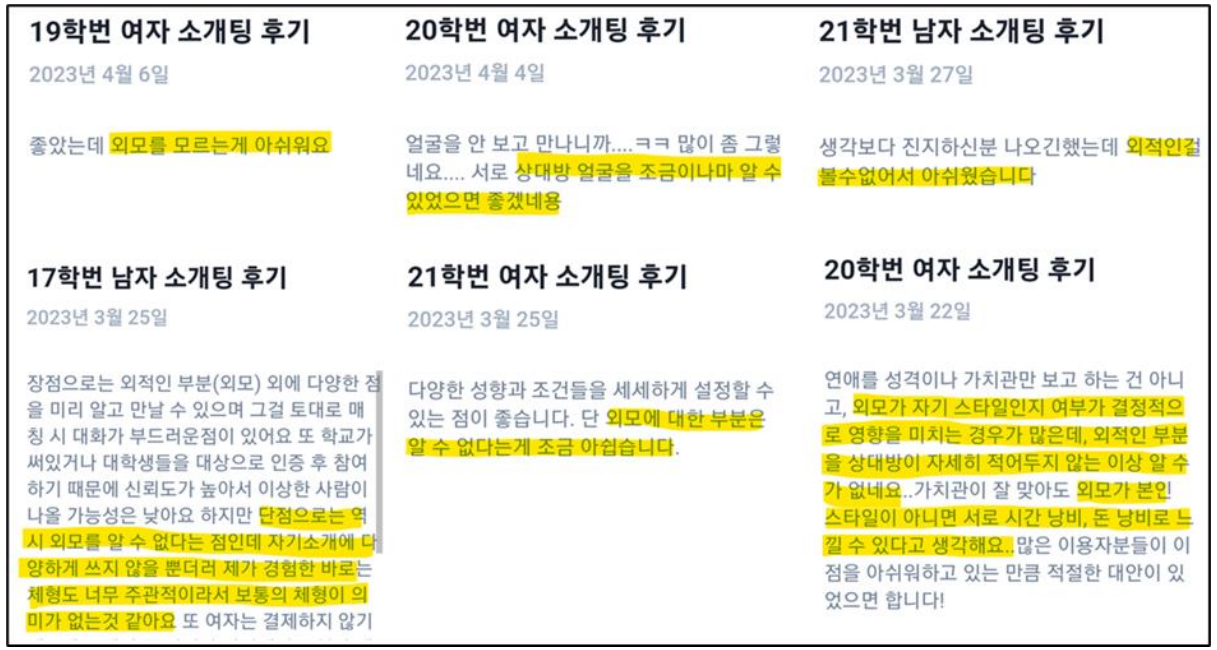
“Get to know the image in a situation where we can’t see the image.”

Why appearance?

- We are not saying that appearance is the most important factor when finding a future partner.
- Appearance is just one factor that people consider when looking for a future partner.
- Other factors are easy to express by writing, but it is not easy to express appearance by writing or quantitive methods.
- It is difficult to describe one's face objectively on one's own.

Is appearance necessarily important?

- People are unconsciously evaluating others' appearance.



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

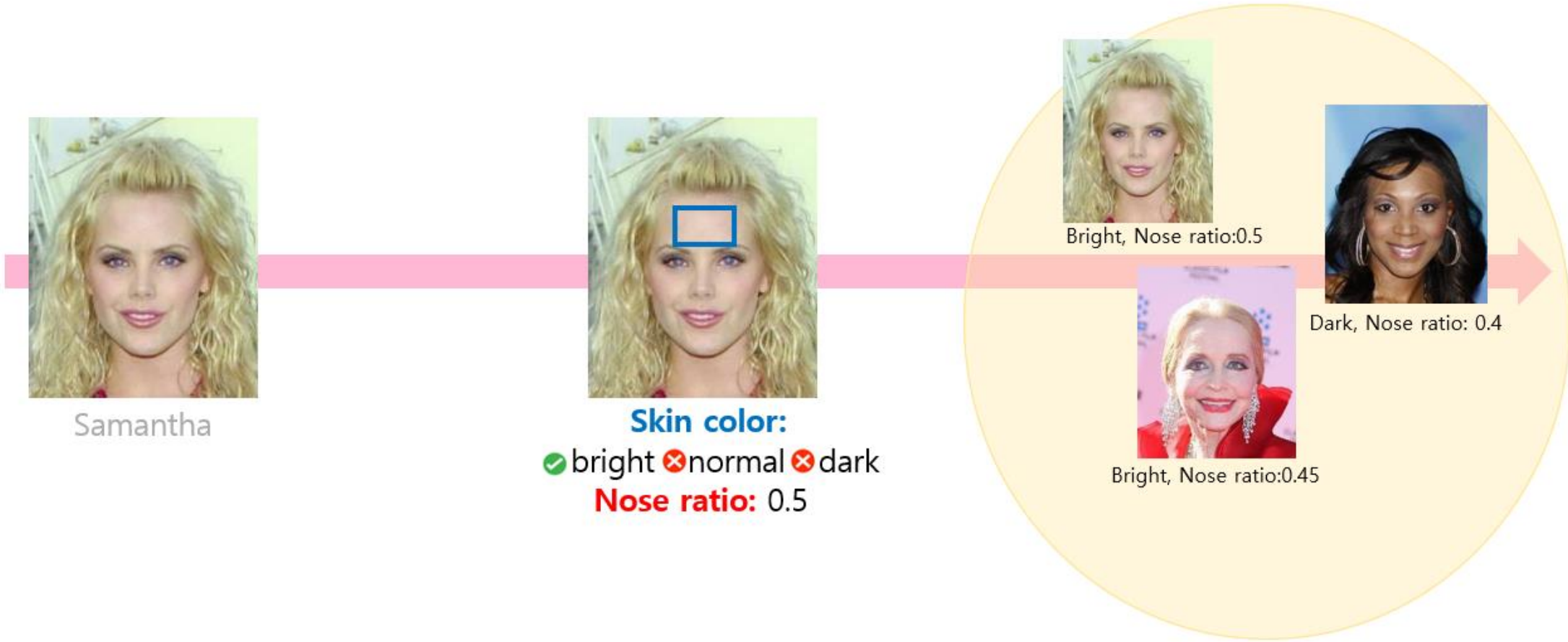
17학번 남자 소개팅 후기
2023년 3월 25일
장점으로는 외적인 부분(외모) 외에 다양한 점을 미리 알고 만날 수 있으며 그걸 토대로 매칭 시 대화가 부드러운점이 있어요 또 학교가 써있거나 대학생들을 대상으로 인증 후 참여하기 때문에 신뢰도가 높아서 이상한 사람이 나올 가능성은 낮아요 하지만 단점으로는 역시 외모를 알 수 없다는 점인데 자기소개에 다양하게 쓰지 않을 뿐더러 제가 경험한 바로는 체형도 너무 주관적이라서 보통의 체형이 의미가 없는것 같아요 또 여자는 결제하지 않기

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.

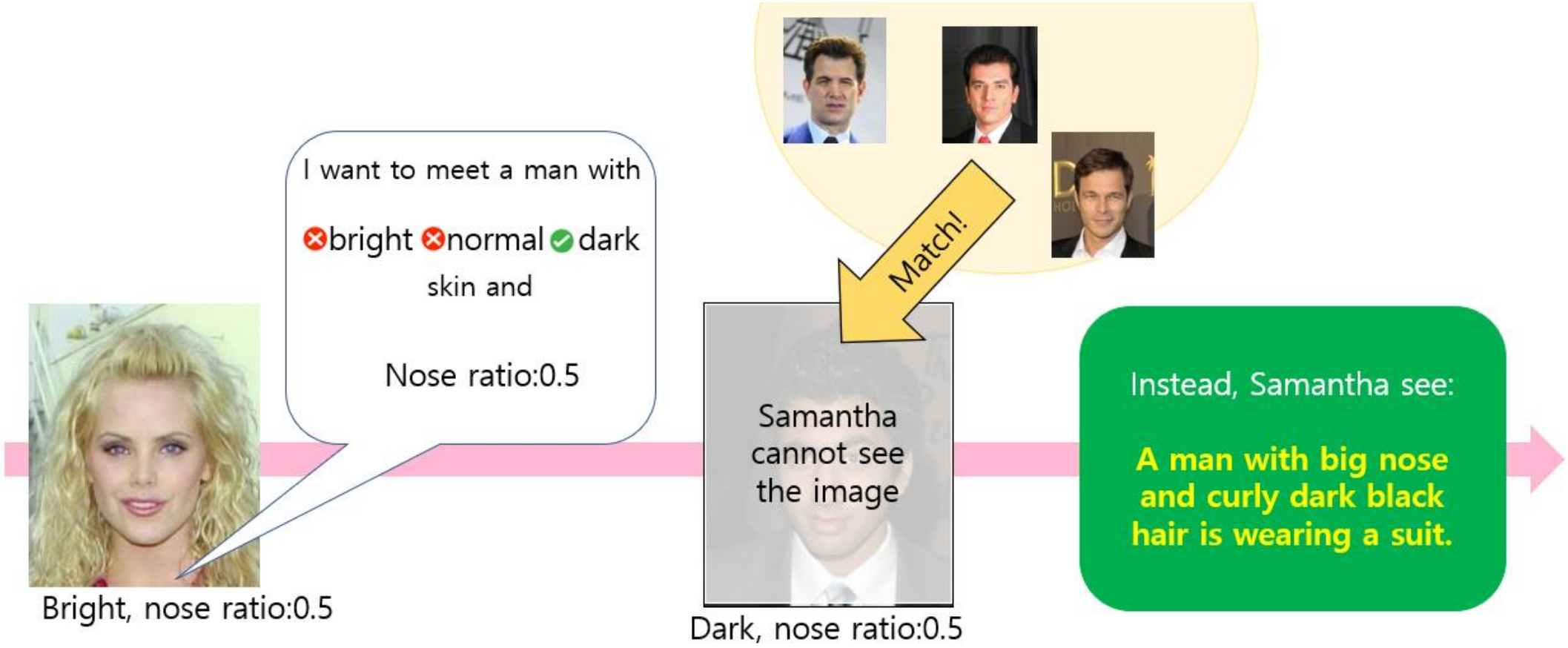
20학번 여자 소개팅 후기
2023년 3월 22일
연애를 성격이나 가치관만 보고 하는 건 아니고, 외모가 자기 스타일인지 여부가 결정적으로 영향을 미치는 경우가 많은데, 외적인 부분을 상대방이 자세히 적어두지 않는 이상 알 수가 없네요..가치관이 잘 맞아도 외모가 본인 스타일이 아니면 서로 시간 낭비, 돈 낭비로 느낄 수 있다고 생각해요..많은 이용자분들이 이 점을 아쉬워하고 있는 만큼 적절한 대안이 있었으면 합니다!

- Reviews of "Yeonpick" saying that it's too bad that they can't know the partner's appearance until they meet in person.

Part1. Introduction: Project Design Sketch



Part1. Introduction: Project Design Sketch



Part2. Previous Works

- **Object Detection**

[1] A Method of Eye and Lip Region Detection using Faster R-CNN Face Image

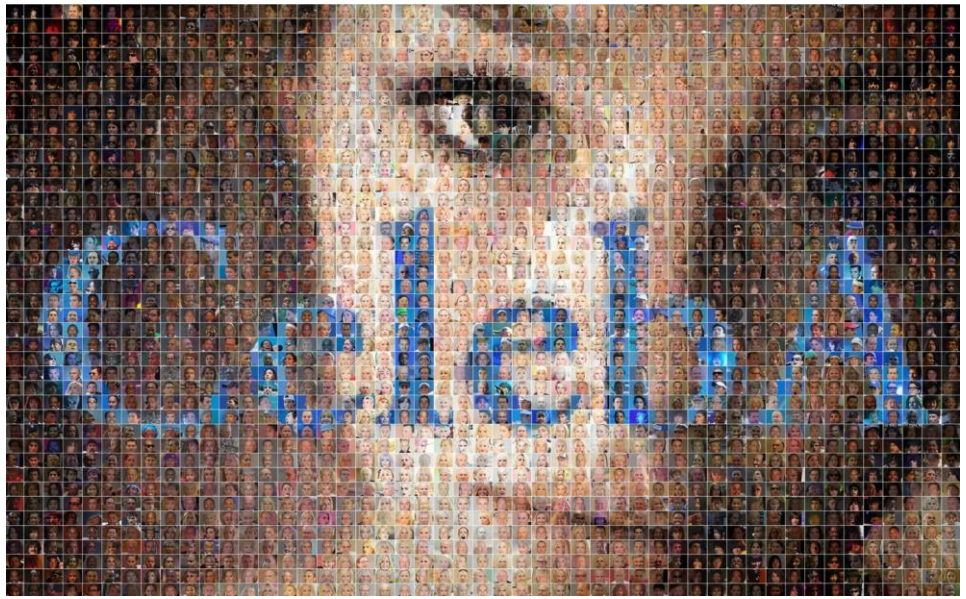
[2] Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks

- **Image Captioning**

[3] Image Caption Generator Using RESNET-LSTM

[4] End-to-End Transformer Based Model for Image Captioning (Yiyu Wang)

Part3. Data



CelebA dataset

- Standardized total 200,000 images (male + female)
- csv files that their characteristics are well organized in

Pre-processing

10,000 women images

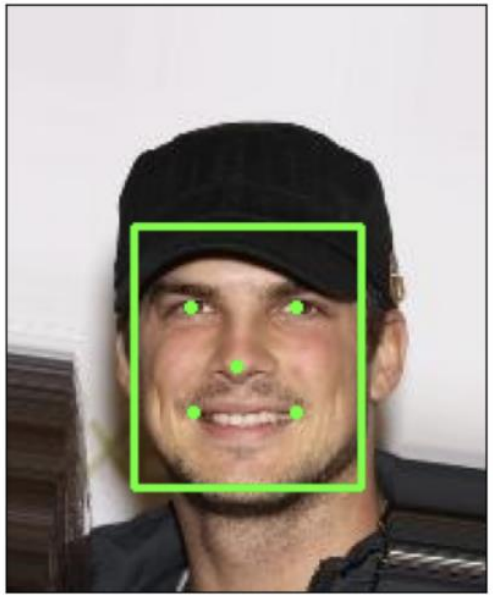
9,429 men images

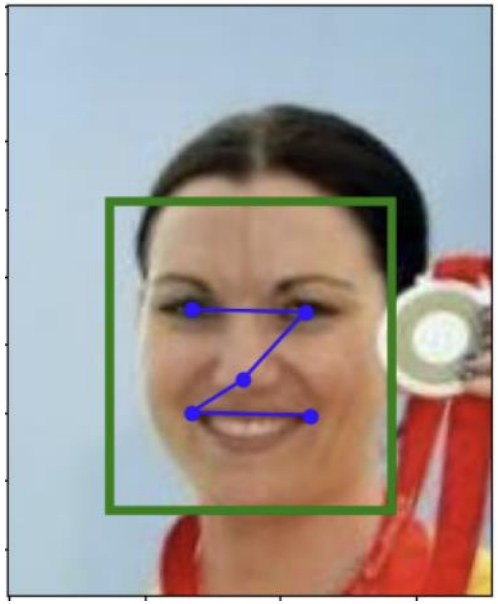
&

csv features equivalent to approximately 20,000 images

Part3. Algorithm – Object Detection

1. Facial Landmark Detection & Extract Facial Information (ratio of nose and lip)
2. Randomly select the image that matches the nose and lip ratio selected by the user





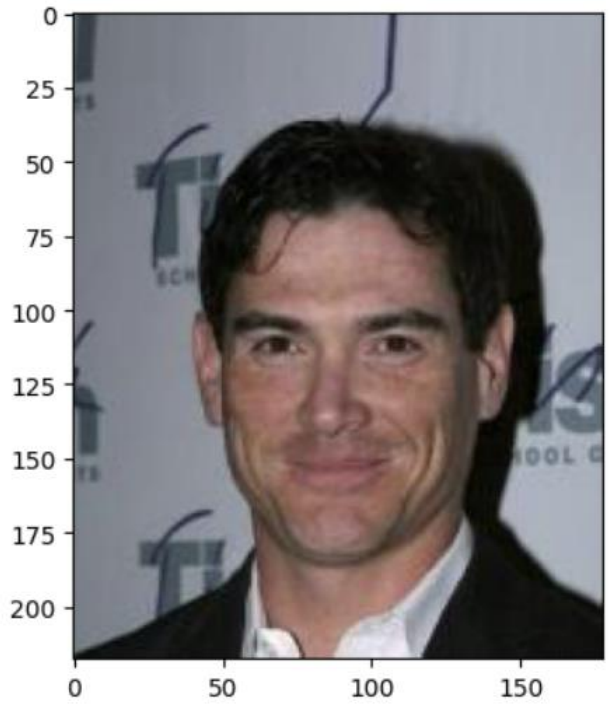
```
[{'box': [47, 82, 84, 97],
  'confidence': 0.9999561309814453,
  'keypoints': {'left_eye': (68, 112),
   'right_eye': (107, 112),
   'nose': (85, 134),
   'mouth_left': (69, 151),
   'mouth_right': (107, 151)}},
 {'ratio': {'face': 8148,
   'mouth': 0.692116278749935,
   'nose': 0.5641025641025641}}]
```

- Mouth ratio = (mouth_right – mouth_left) / box_w
- Nose ratio
  1. Connect the left_eye and right_eye in a straight line
  2. Find the distance between the point of the nose and the straight line
  3. Divide the value to box_h

3. Extract skin color by cropping the image and calculating Luminance, and randomly select the image of skin color that the user desires.

Part3. Algorithm - Image Captioning

Generate captions for the selected image

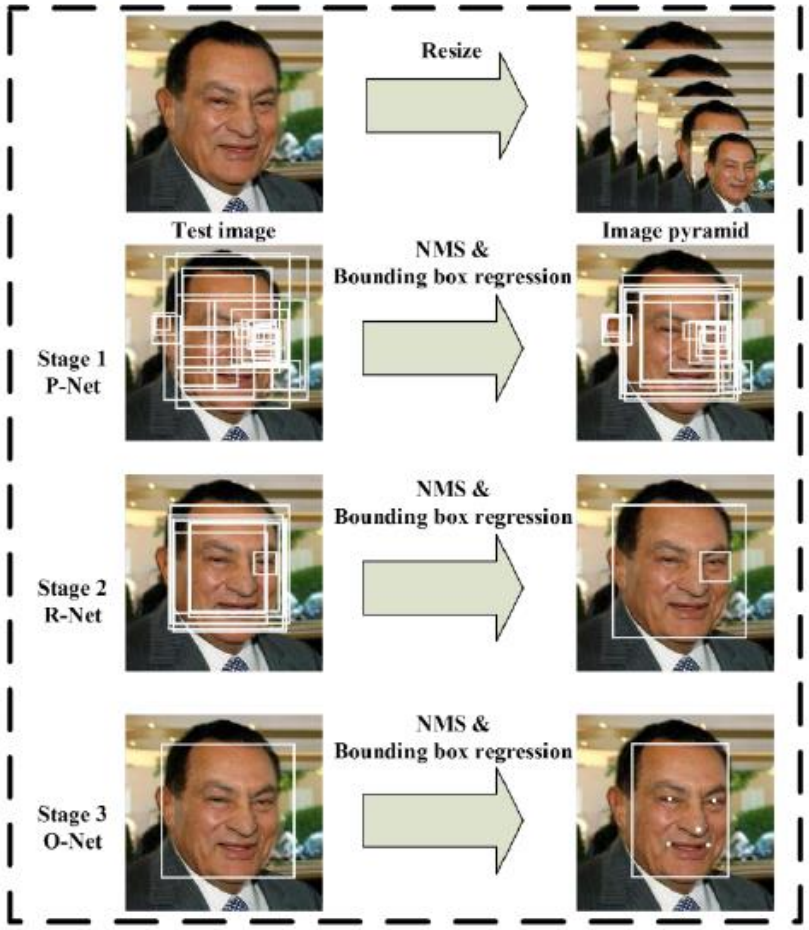


<start> a man in a black shirt and a woman in a black shirt and a woman in a black

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

Part4. Method - Yoonseon

Image Detection - MTCNN



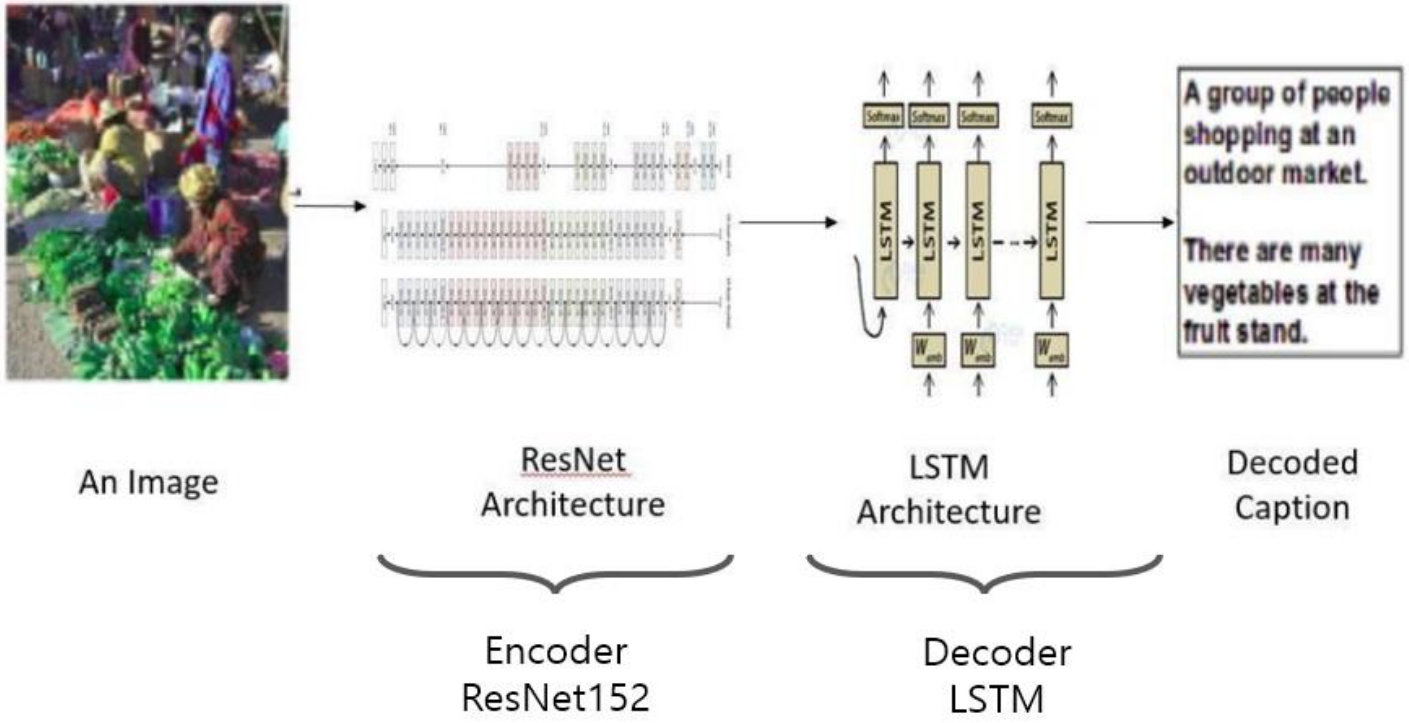
- To better detect faces of various sizes, the input image is resized to multiple scales to create an image pyramid.
- It is a network that finds faces in images and is a network consisting of Conv layers without FC layers.
- It is almost similar to P-Net, and FC layer was added at the end.
- It's a network that finds face landmarks. Through several Conv layers and FC layers, three types of outputs are released: face classification, bbox regression, and face landmark localization.

<Training Data>

- Negatives: Regions that IoU ratio less than 0.3 to any GT faces
- Positives: IoU above 0.65 to GT faces
- Part faces: IoU between 0.4 and 0.65 to GT faces
- Landmark faces: faces labeled 5 landmarks' positions

Part4. Method - Yoonseon

Image Captioning – ResNet152+LSTM



<Problem>

- We couldn't manually create caption data for all the data of CelebA.

<Solution>

- Let's pre-train the model with Flickr8k dataset.

<Training Data>

- Flickr8k dataset

<Hyperparameter>

- Loss: Cross entropy
- Epoch: 5
- Learning rate: 0.001

Part4. Method - Yeeun

Image Detection – Faster R-CNN

torchvision.models.detection.fasterrcnn_resnet50_fpn

```
FasterRCNN(
  (transform): GeneralizedRCNNTransform(
      Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
      Resize(min_size=(800,), max_size=1333, mode='bilinear')
  )
  (backbone): BackboneWithFPN(
    (body): IntermediateLayerGetter(
      (conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
      (bn1): FrozenBatchNorm2d(64)
      (relu): ReLU(inplace=True)
      (maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
      (layer1): Sequential(
        (0): Bottleneck(
          (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn1): FrozenBatchNorm2d(64)
          (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
          (bn2): FrozenBatchNorm2d(64)
          (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (bn3): FrozenBatchNorm2d(256)
          (relu): ReLU(inplace=True)
          (downsample): Sequential(
            (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
            (1): FrozenBatchNorm2d(256)
          )
```

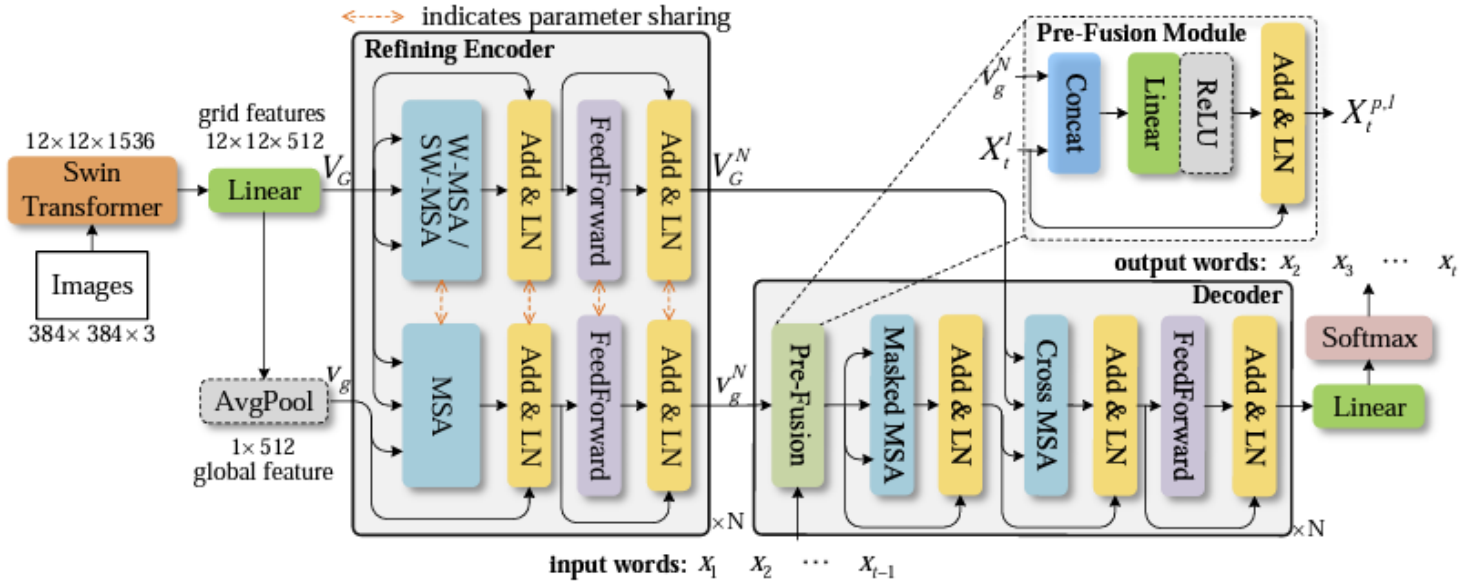
-
-
-

<Hyperparameters>

- Optimizer: SGD
- Learning rate: 0.005
- Momentum: 0.9
- Weight decay: 0.0005
- Epochs: 10

Part4. Method - Yeeun

Image Captioning – Transformer-based



- Adopt SwinTransformer to replace Faster R-CNN as the backbone encoder to extract grid-level features from given images.
- 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.
- 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.

<Training Data>

- Flickr8k dataset

<Hyperparameter>

- Optimizer: Adam
- Loss: Cross Entrophy
- Epoch: 1

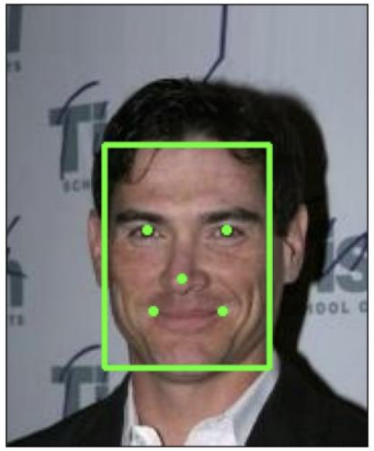
- applied EfficientNetB2 for image feature extractor (instead of SwinTransformer) and skip refining encoder

Part4. Result – Yoonseon (Object detection – MTCNN)

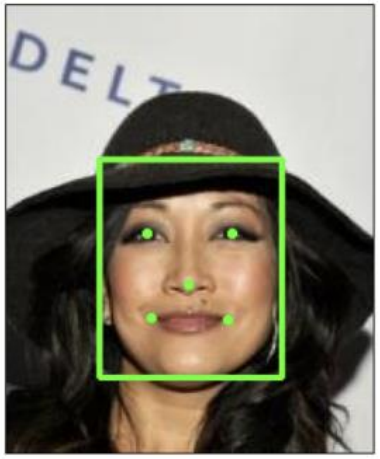
- User enter

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

- Detected Image & Information



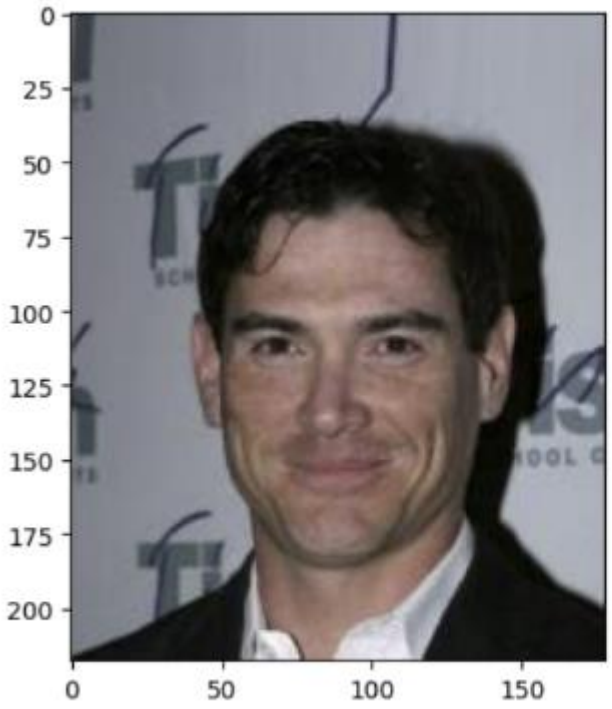
```
[{'box': [48, 69, 81, 110], 'confidence': 0.9994163513183594,
'keypoints': {'left_eye': (69, 111), 'right_eye': (108, 111), 'nose': (86, 135), 'mouth_left': (72, 151), 'mouth_right': (106, 151)}},
{'ratio': {'face': 8910, 'mouth': 0.6963006741828925, 'nose': 0.6153846153846154}}]
```



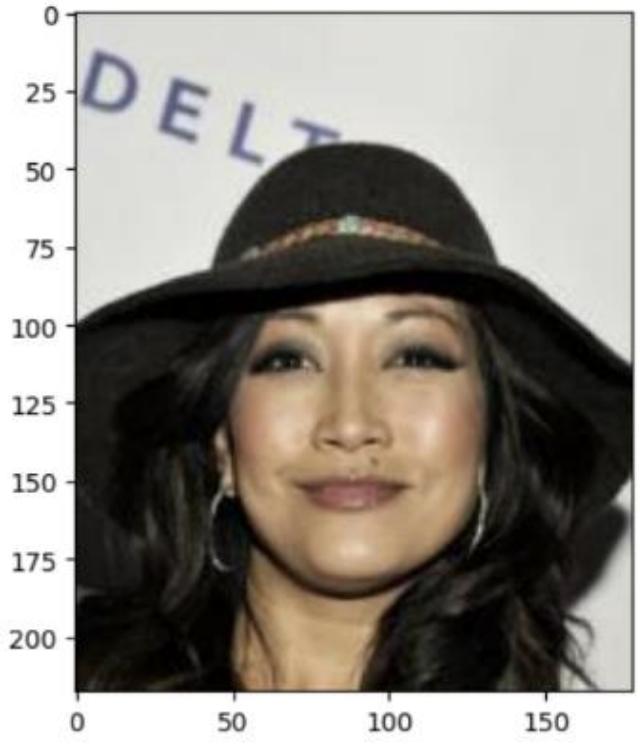
```
[{'box': [45, 75, 88, 106], 'confidence': 0.9997329115867615,
'keypoints': {'left_eye': (68, 111), 'right_eye': (109, 111), 'nose': (88, 136), 'mouth_left': (70, 152), 'mouth_right': (107, 153)}},
{'ratio': {'face': 9328, 'mouth': 0.6708396428702473, 'nose': 0.6097560975609756}}]
```

Part4. Result – Yoonseon (Image Captioning – Resnet152-LSTM)

- Generating captions of the selected image

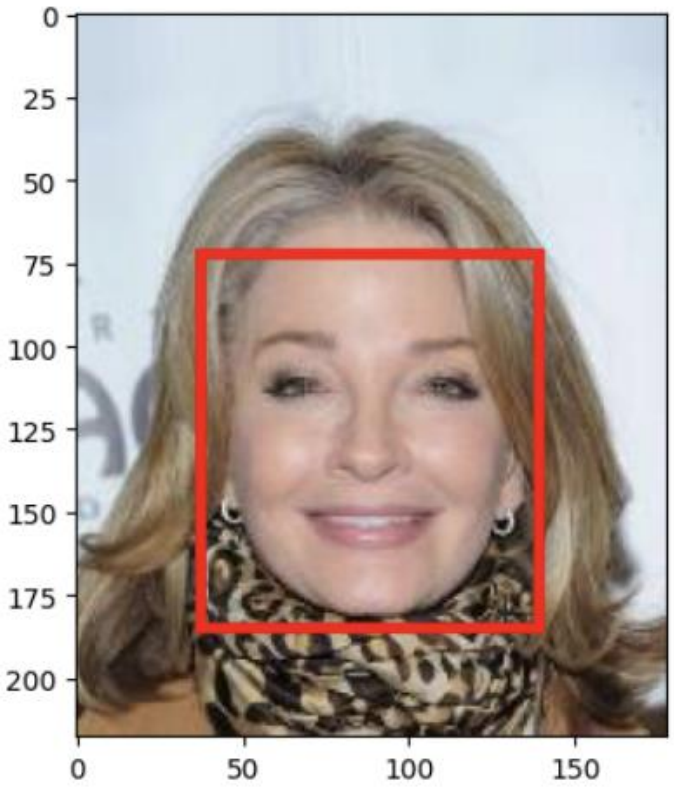


<start> a man in a black shirt and a woman in a black shirt and a woman in a black



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

Part4. Result – Yeeun (Object detection – Faster R-CNN)



1. Predicted result by Faster R-CNN

```
cutimage= crop_image(image_index)

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


2. Crop the image (forehead / cheek)

```
Avg Value
R :  164.68077601410934
G :  179.6851851851852
B :  211.89770723104056
Luminance:  178.82099188712522
178.82099188712522
```

```
luminance= 0.2126 * R_avg + 0.7152 * G_avg + 0.0722 * B_avg
```

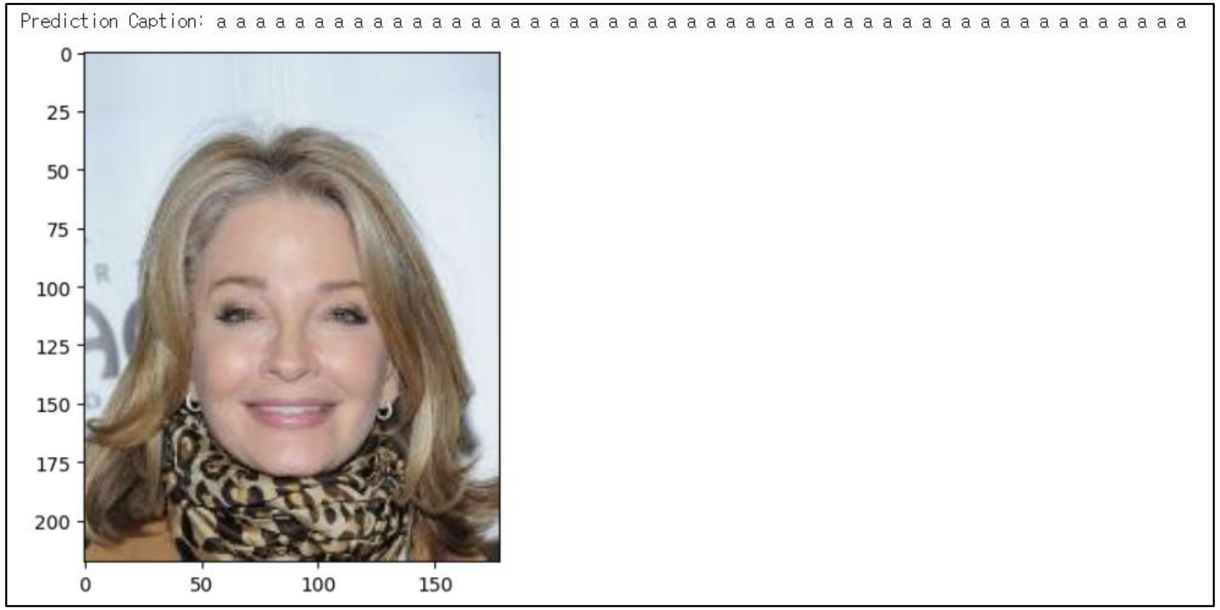
3. Calculate Luminance and Label based on standard

⇩

Label: bright

User: I want to meet a man with Dark Skin
Algorithm: (Matches with a man who has dark skin and wants to meet a woman with bright skin)

Part4. Result – Yeeun (Image Captioning – Transformer-based)



➡ Inaccurate result by small number of Epochs?

Part5. Conclusion

Object detection

- **MTCNN**
- **Faster R-CNN**

→

- Both can detect face
- Can be used for calculating the luminance of skin color / mouse, nose ratio
- Accurate result

Image captioning

- **ResNet152-LSTM**
- **Transformer-based**

→

- Both can generate captions of Image
- Inaccurate result

Thank you!