

# YOLO를 이용한 내 눈 관상 찾기

내가 왕이 될 상인가

강민영 오상수 정지희

# INDEX

- 기획
- 개발 디자인
- 발표
- 향후 계획

# 기획

## Outline

01

## »» 개발 개요

02

03

04

# 눈 = 얼굴에서 중요하다 여겨지는 부위

- 코로나 이후 가장 드러나는 부위
- 사람을 인식할 때 가장 큰 비율을 차지하는 부위
- 외부로부터 받는 정보의 약 70%가 시각

## »» 눈과 관련된 것을 만들어보자

01

## &gt;&gt; 개발 개요

02

03

04

## 딥러닝을 이용한 눈 관상 분석



- 눈 관상 판별  
: Precision(정밀도) / recall(재현율)
- 눈 검출
- 길상, 평상, 흉상 판별

## 기능

1. 실시간 확인 가능
2. 화면 캡처 및 저장 가능

## 사용법

: 얼굴 중점으로 눈은 카메라를 정면으로 바라본다

# 개발 디자인

## Design

01

## >> 개발 환경

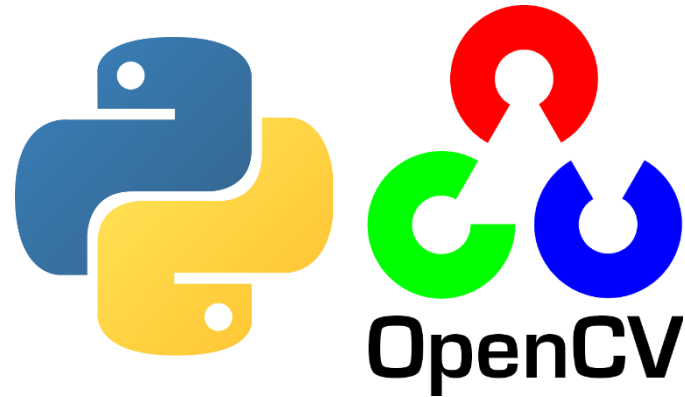
02

03

04

colab

YOLO



01

## >> Class List

02

- 약 22개의 관상 -> 6개로 줄임

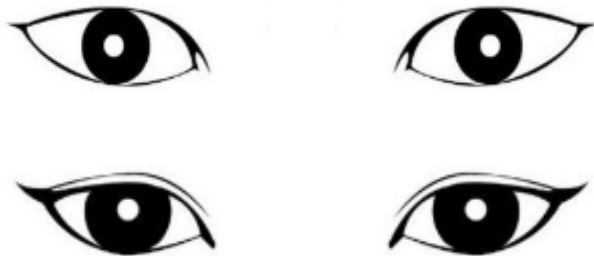
03

dragon ( good )



04

lion ( good )



cow ( good )



crane ( normal )



sparrow ( normal )



Umm ( bad )





01

02

03

04

## » Data set 및 cfg 설정

- Learning\_rate=0.001
- max\_batches = 12000 ( 6개 class \*2000 )
- 20개의 train set (이미지 한 장 당 다양한 데이터를 포함)
- 2개의 test set



01

## >> Code

02

- Yolo Train code

```
!./darknet detector train custom/custom_data.data custom/custom-train-yolo.cfg  
weights/darknet53.conv.74 -dont_show
```

03

04

- Yolo 학습시 기존 진행했던 학습에서부터 이어서 training 가능 / 추가 학습 불가능

```
!./darknet detector train custom/custom_data.data custom/custom-train-yolo.cfg  
backup/custom-train-yolo_4001.weights -dont_show
```

- Colab max runtime 12시간으로 백업을 해두지 않으면 학습한 yolo weight 초기화

### 1) weight 파일을 google 드라이브에 backup

```
!cp -r "/content/darknet/backup" "/content/drive/MyDrive/darknet/"
```

### 2) colab 방지 코드

F12 - console

```
function ConnectButton(){  
  console.log("Connect pushed");  
  document.querySelector("#top-toolbar >  
    colab-connect-button").shadowRoot.querySelector("#connect").click()  
}  
setInterval(ConnectButton,60000);
```

01

## >> Data Train

02

Iteration

loss average

1:	1976.273071,	1976.273071	avg loss, 0.000000 rate, 4.445358 seconds, 32 images
Loaded: 0.000061 seconds			
2021:	0.200304,	0.188936	avg loss, 0.001000 rate, 1.853482 seconds, 64672 images
Loaded: 0.000064 seconds			
5000:	0.047292,	0.056249	avg loss, 0.001000 rate, 3.335451 seconds, 160000 images
10000:	0.015087,	0.023582	avg loss, 0.000100 rate, 2.499065 seconds, 320000 images
12000:	0.025911,	0.021309	avg loss, 0.000010 rate, 4.096377 seconds, 384000 images

03

04

- Iteration이 증가할 수록 loss avg 값 감소
- loss avg가 0.25이하로 도출되도록 하는 게 목표
- 학습시간 : 1000회당 약 50분, 12000회 약 10시간

01

## » 모델 성능 평가

02

03

04

```
detections_count = 9, unique_truth_count = 6
class_id = 0, name = dragon, ap = 100.00%      (TP = 1, FP = 0)
class_id = 1, name = lion, ap = 0.00%          (TP = 1, FP = 1)
class_id = 2, name = cow, ap = 100.00%         (TP = 1, FP = 0)
class_id = 3, name = crane, ap = 100.00%       (TP = 1, FP = 0)
class_id = 4, name = sparrow, ap = 0.00%       (TP = 0, FP = 0)
class_id = 5, name = Umm, ap = 0.00%          (TP = 0, FP = 0)
```

for thresh = 0.25, precision = 0.80, recall = 0.67, F1-score = 0.73

for thresh = 0.25, TP = 4, FP = 1, FN = 2, average IoU = 72.16 %

- Precision (정밀도) : 0.80 높은 편
- Recall(재현율) : 0.67 낮은 편
- F1 score (Precision과 Recall의 조화평균) : 0.73

- Recall이 낮은 이유

- 1) class 개수를 줄이기 위해 여러 관상을 임의로 묶어서 결과를 도출함( 사자상 + 호랑이상 → lion상 )
- 2) data sample 개수 부족
- 3) 다수의 사진 변수 요인 (보정, 눈각도 등)
- 4) test set 개수 부족

01

02

03

04

## >> Python code

- 길상 / 평상 / 흉상 판별

```
if names[i]=='dragon' or names[i]=='lion' or names[i]=='cow':  
    cv2.putText(img, 'Good', (x, y+h+12), font, 1, color, 2)  
    if names[i]=='crane' or names[i]=='sparrow':  
        cv2.putText(img, 'Normal', (x, y+h+12), font, 1, color, 2)  
    if names[i]=='Umm':  
        cv2.putText(img, 'Bad', (x, y+h+12), font, 1, color, 2)
```

- 판독 영상 캡처 후 자동 저장

```
if cv2.waitKey(1) == ord('c'): # 캡처(화면 멈춤)  
    cv2.IMREAD_UNCHANGED  
    cv2.imwrite("image/cap01"+"%.png",img) # 캡처 저장  
    cv2.waitKey(0)  
  
...  
if cv2.waitKey(1) & 0xFF == ord( ' q ' ): # 화면 멈춤 해제 (1회 더 누르면 종료)  
    break
```

DEMO

Play sample

01

## >> Image demo

02

03

04

05

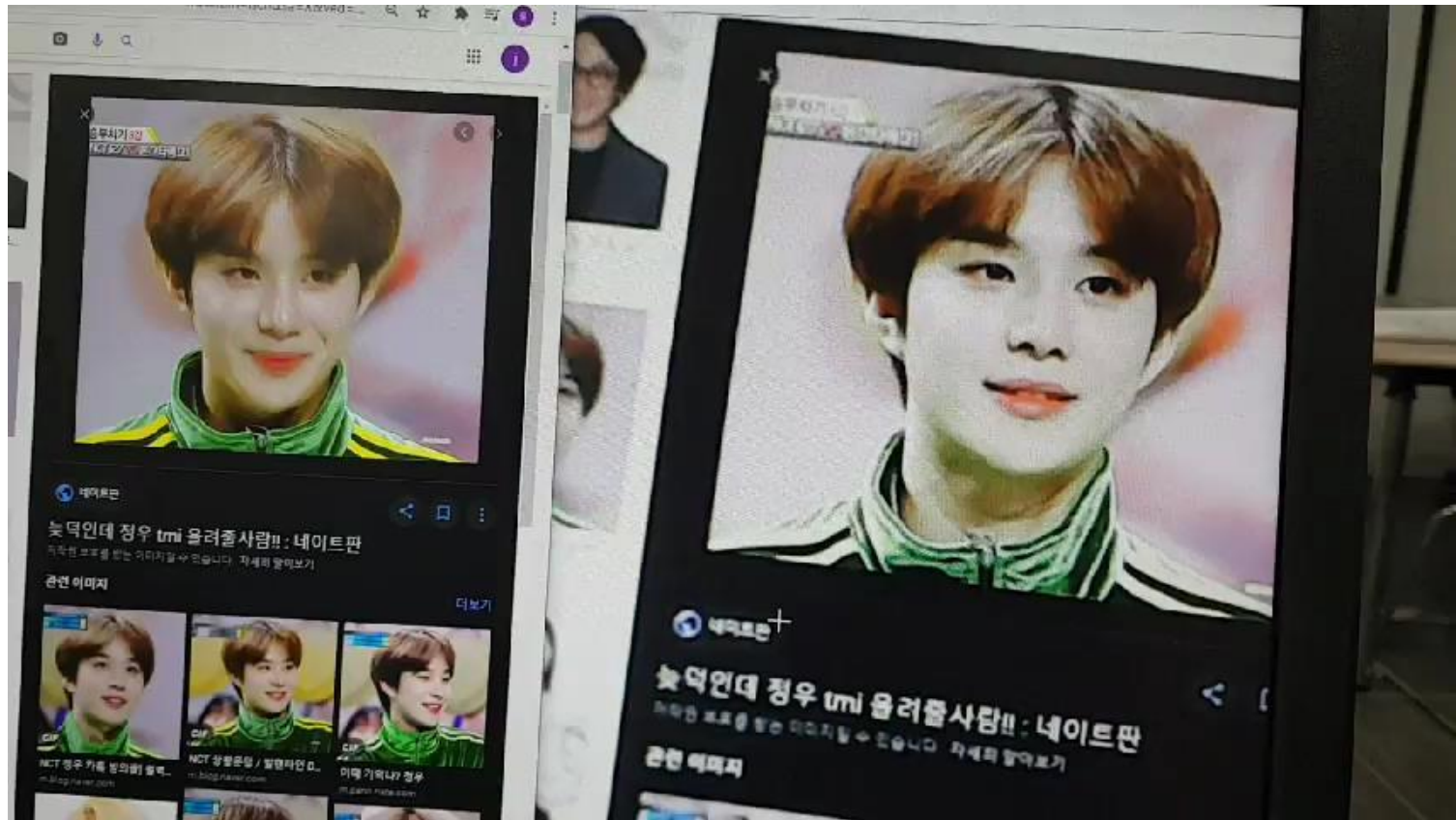


```
...
Total BFLOPS 65.326
Allocate additional workspace_size = 52.43 MB
Loading weights from backup/custom-train-yolo_final.weights...
seen 64
Done!
data/cc.jpg: Predicted in 41.034000 milli-seconds.
crane: 90%
Unable to init server: Could not connect: Connection refused

(predictions:39883): Gtk-WARNING **: 08:54:10.892: cannot open display:
```

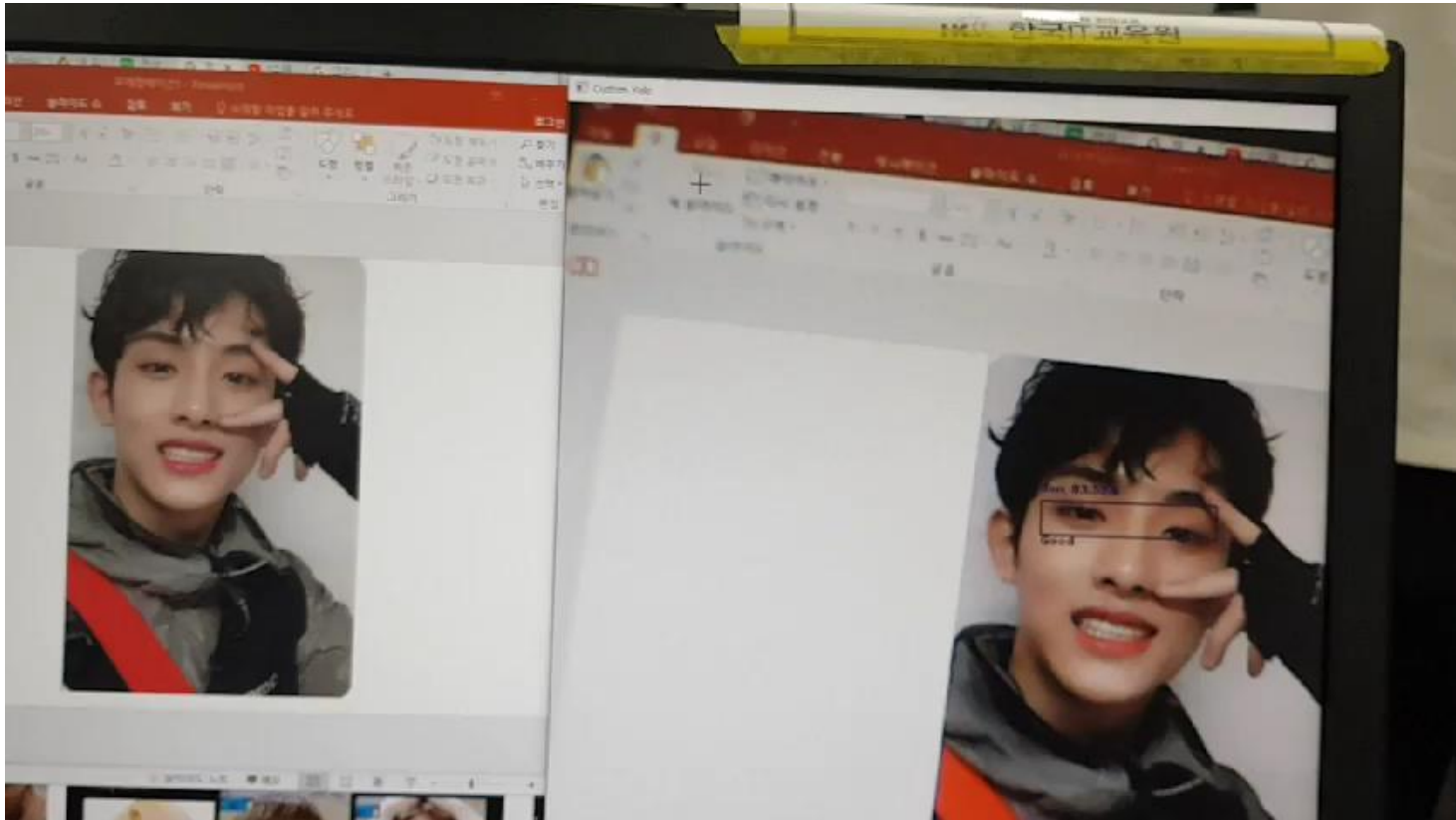


## » Video demo





## » Video demo



향후 계획

01

02

03

04

## »» 향후 계획

안드로이드 APP 구현



안드로이드 학습

결과값 개선



정확도 높은 사진 추가  
추가 training

## »» 응용 할 수 있는 분야

- 백내장 및 안구 질환 관련 프로그램
- 코로나19로 인한 마스크 쓴 얼굴인식 가능 프로그램

Thank you