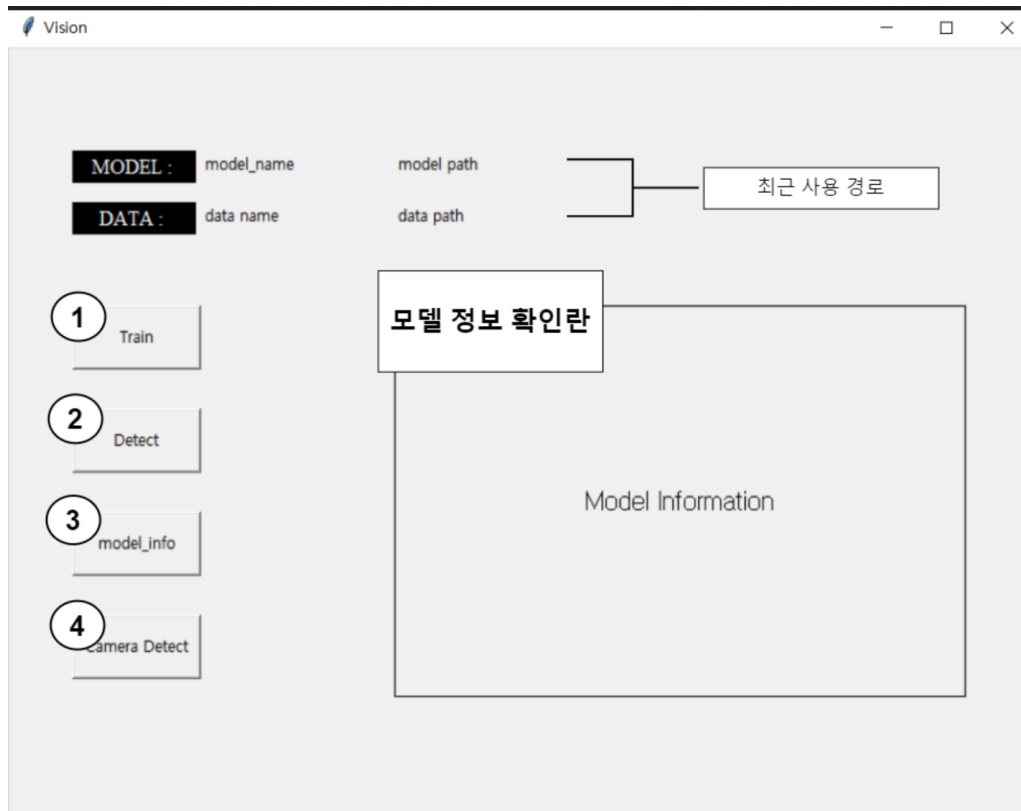


# Darknet UI guide



## 1. Train 버튼

- 모델 생성

## 2. Detect 버튼

- 폴더 내부의 이미지 detect

## 3. 모델 정보 버튼

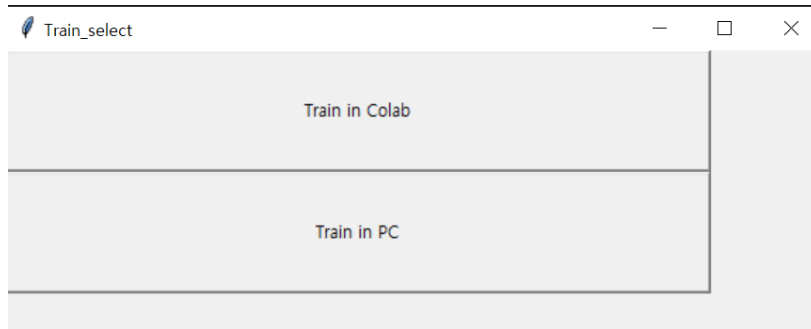
- 선택한 모델의 정보를 모델 정보 확인란에 생성

## 4. 실시간 카메라 Detect버튼

- 연결된 카메라를 이용한 실시간 카메라 Detect

# 1. Train

Train in Colab, Train in PC 중 택일



## 1. Train in Colab

-Google Colab에서 Train

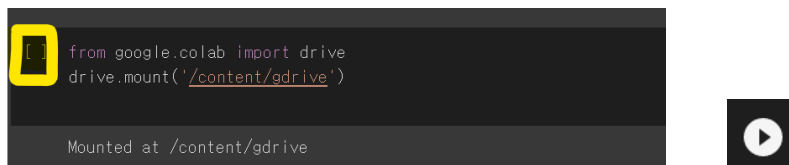
클릭시 3가지 페이지가 열립니다.

<https://kimcharless.tistory.com/1>

↳ 해당 페이지를 참고하여 <https://roboflow.com/> 이곳에서 Custom Data set을 만들고

[https://colab.research.google.com/drive/1Vha1JowozqywCfdRUzAh2PSgb\\_uuZvy\\_?authuser=1](https://colab.research.google.com/drive/1Vha1JowozqywCfdRUzAh2PSgb_uuZvy_?authuser=1)

구글에 로그인 후 해당 코랩에서 작성되어있는대로 따라하시면 됩니다.



노란색 박스 우측에 있는 Play버튼을 하나씩 누르시거나, Shift+Enter을 누르면 하나가 실행되고 다음으로 이동합니다.

모두 완료 후

Obj.cfg , obj.names, yolov4-obj.cfg ,information.txt, backup폴더의 last\_weights를

darknet폴더의 models안에 모델이름으로 폴더 생성 후 모두 넣으면 됩니다.

## 2. Train in PC

-PC 자체 그래픽카드를 사용해 Train

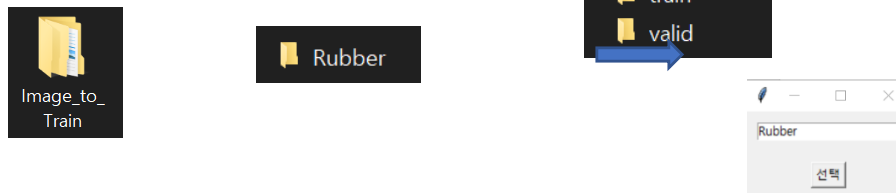
Name of Model : 모델이름을 입력

만든사람 이름: 만든사람의 이름을 입력

Name of Data: dark/Image\_to\_train 폴더에 train할 이미지를 넣습니다.

Train할 이미지는 roboflow에서 <https://roboflow.com/> 작성하시면 됩니다.

<https://kimcharless.tistory.com/1> 해당 페이지 참고



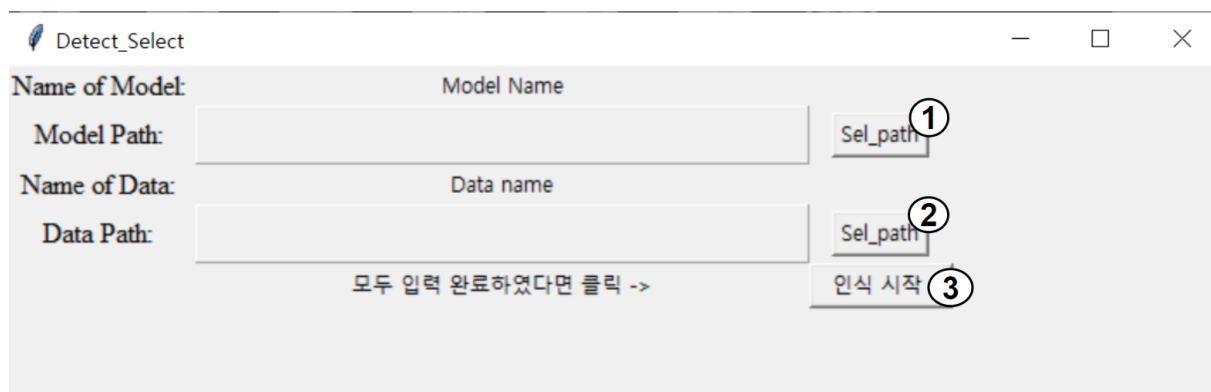
Image\_to\_Train 내부의 폴더명을 기준으로 Sel\_path을 눌렀을때 선택할 수 있습니다.

모두 입력을 완료했다면 학습시작을 누릅니다.

Obj.names,obj.data,yolov4-obj.cfg,information.txt폴더가 생성되며, models폴더의 models.json파일이 업데이트 됩니다.

새로운 창이 열리면서 Train이 시작되고, 일정수준 Train이 되면 darknet/models/이름 폴더의 weights가 저장 됩니다.

## 2. Detect



버튼1.

Models폴더 내부의 사용할 모델을 선택합니다

버튼2.

경로 기반으로 detect할 데이터의 폴더를 선택합니다

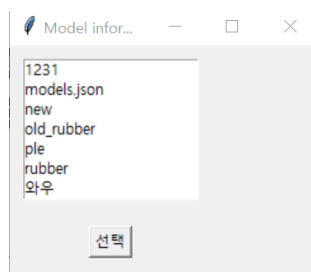
버튼3.

새 창이 열리며 detect가 시작됩니다.

결과물은 darknet/temp폴더에 저장됩니다.

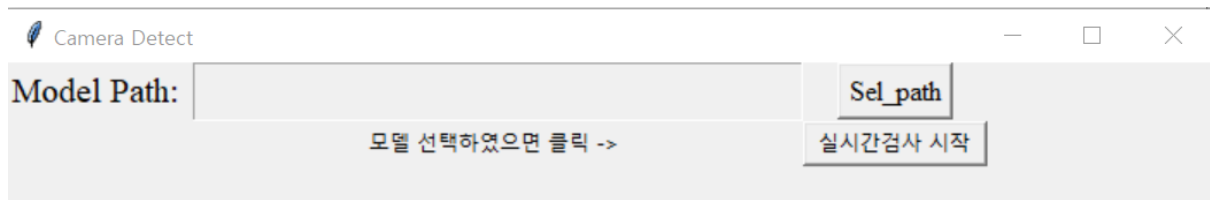
Darknet/darknet\_images.py파일의 229번째 줄에서 원하는 파일위치를 설정할 수 있습니다.

## 3. model\_info



models폴더 각각의 obj.names , information.txt를 기반으로 모델의 정보를 볼 수 있습니다.

## 4. Camera Detect



basler카메라와 연동시켜 선택한 모델을 기반으로 실시간 detect를 수행합니다.

Sel\_path를 클릭시 models폴더안의 사용할 모델을 선택 가능하고, 실시간검사 시작을 클릭시 darknet/CAM/CCAM.py 파일이 실행되며 실시간검사가 시작됩니다.

## 그 외 주의점

\*Darknet/models/models.json에서 모든 모델의 정보를 확인 가능합니다.

프로그램을 시작할때 생성되며, 새로 모델을 만들시 업데이트됩니다.

\*Train시에 이미지는 반드시 Image\_to\_train폴더 안에 존재해야 하며, 폴더 안에 train과 valid폴더가 존재해야 합니다. Train시에 필요한 이미지 리스트 txt는 train시작시 자동 생성됩니다.