



COVID19

Mask Detection

최종발표

구형석 권영진
이노아 임동휘
임현우

시중 제품과의 차별점?



+코로나극복
비대면 얼굴인식 발열 체크 열화상 카
메라 온도 측정
2,500,000원

- 제품별로 몇 백 만원 수준
- 시스템 구축까지 하려면 최대 700만원

정확하게 어떤 모델을 쓰는지, 얼마만큼의 정확성을 가지는지는 공개 X

Checking Feedback

시중 제품과의 차별점?

개인 노트북과 웹 캠만으로도 구동 가능하다

해당 제품을 더 다양한 주체들이 사용 가능하도록 할 수 있다

Ex. 일회성 행사를 개최할 때, 예산이 부족한 경우

시중 제품들과 목표하고 있는 것이 다르다

Checking Feedback

인터넷상 레퍼런스들? 새로운 모델 개발 목표? No

Project for Studying

- 4200장의 자체 데이터셋 구축
- 중간 발표 이후, 다양한 실험 환경과 모델 시도
- 3 label mask detector(Detecting correct, incorrect, no mask)

1. Detecting Correct, Incorrect, No Mask

최초 시도, 필요한 이유?

현직 초등학교 보건교사, “초등학생 마스크 교육에 사용하고 싶다”

자율주행 로봇 + Mask Detector, “마스크 착용 필수 시설에 도입”

2. 4200장의 correct, incorrect, no mask 자체 데이터셋 구축

목표: 딥러닝 이미지 관련 프로젝트의 처음부터 끝까지 경험

데이터셋 제작 -> 모델 학습 -> 시연 -> 모델 개선 -> 서버에 올리는 것까지

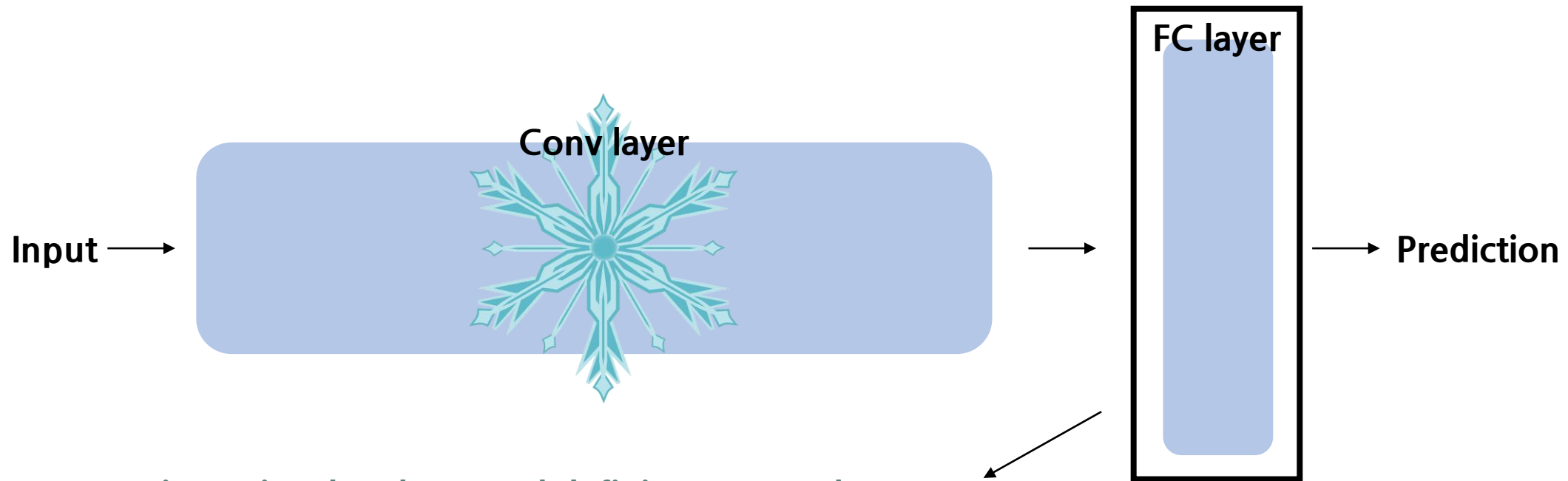
1. 99% Accuracy Model
2. Employing to image, video, and real-time video
3. Training Image classification & Object Detection model
4. Web Application Development

Transfer Learning

Transfer-Learning by using pre-trained MobilenetV2

Weight: Trained by "Imagenet"

Number of Output label = 1000(existed) => 3(correctly, incorrectly, no mask)



Removing existed FC layer and defining new FC layer
While training, freeze Conv layer and only update FC layer

Transfer Learning - MobilnetV2

Input	Conv Layer	FC Layer	Accuracy
4,200 images (correct, incorrect, no)	Pre-trained weight (Freeze)	Newly Defined	<div><div>precisionrecall</div><div>cor_mask0.990.98</div><div>incor_mask0.980.99</div><div>nomask0.991.00</div></div>
	Pre-trained weight (Not-Freeze)	Newly Defined	<div><div>precisionrecall</div><div>cor_mask0.991.00</div><div>incor_mask1.001.00</div><div>nomask0.990.99</div></div>
	Not-trained weight (Not-Freeze)	Newly Defined	<u>Not-trained</u>

Transfer Learning - InceptionV3

Input	Conv Layer	FC Layer	Accuracy
4,200 images (correct, incorrect, no)	Pre-trained weight (Freeze)	Newly Defined	<div><div>precision</div><div>recall</div><div>cor_mask0.980.97</div><div>incor_mask0.970.97</div><div>nomask0.990.99</div></div>
	Pre-trained weight (Not-Freeze)	Newly Defined	<div><div>precision</div><div>recall</div><div>cor_mask1.001.00</div><div>incor_mask0.971.00</div><div>nomask1.000.97</div></div>
	Not-trained weight (Not-Freeze)	Newly Defined	<div><div>precision</div><div>recall</div><div>cor_mask1.000.91</div><div>incor_mask0.911.00</div><div>nomask1.001.00</div></div>

Total Parameters: 22,065,443 / Model Size: 89MB

Compressing the model - tensorflow lite

Tensorflow Lite: a lightweight library for deploying models on mobile and embedded devices

The TensorFlow Lite converter converts TensorFlow models into an efficient form

How it works



Pick a model

Pick a new model or retrain an existing one.



Convert

Convert a TensorFlow model into a compressed flat buffer with the TensorFlow Lite Converter.



Deploy

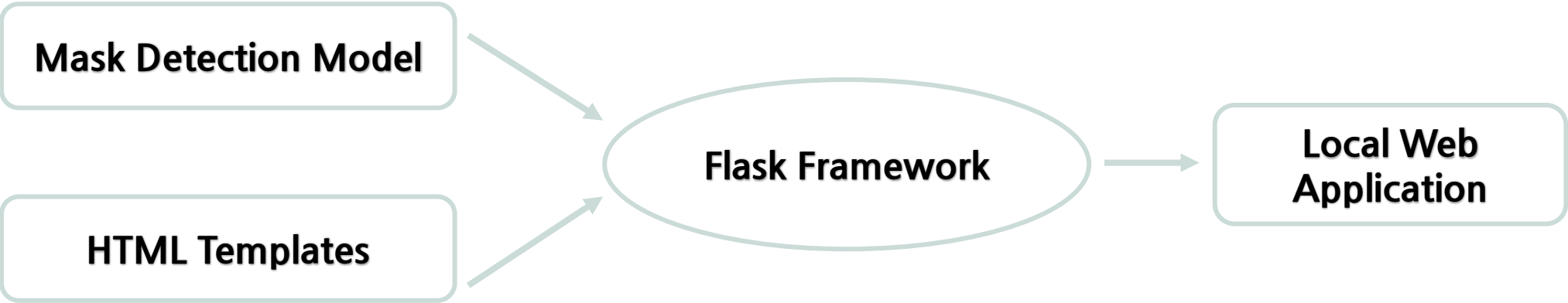
Take the compressed .tflite file and load it into a mobile or embedded device.



Optimize

Quantize by converting 32-bit floats to more efficient 8-bit integers or run on GPU.

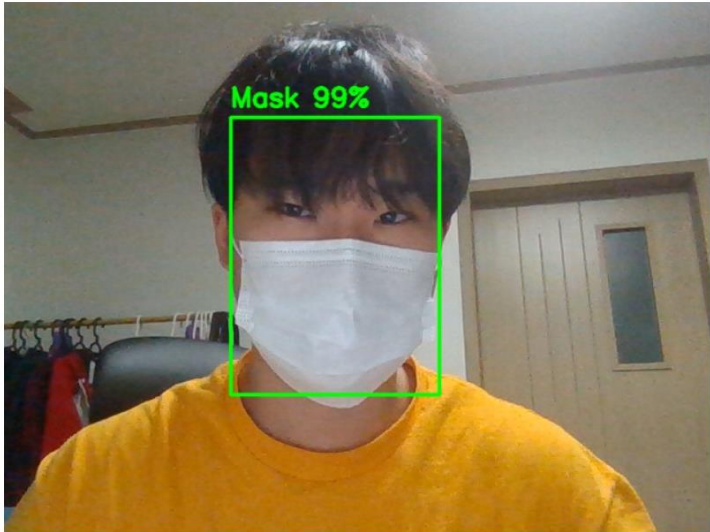
InceptionV3 model (89MB) => InceptionV3 tf lite model (86MB)...



Web Application Development



실시간 마스크 탐지가 작동중입니다



* 로컬 서버에서 작동 중입니다

Web Application Development



Object Detection

Image Classification model

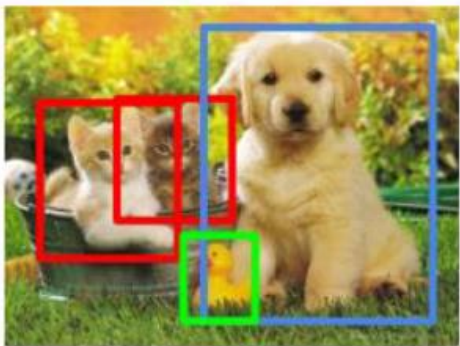
Object Detection model
: Region of proposal + Image Classification

2-stage: RCNN, Fast RCNN, Faster RCNN
1-stage: SSD, Yolo

Image Segmentation model



CAT

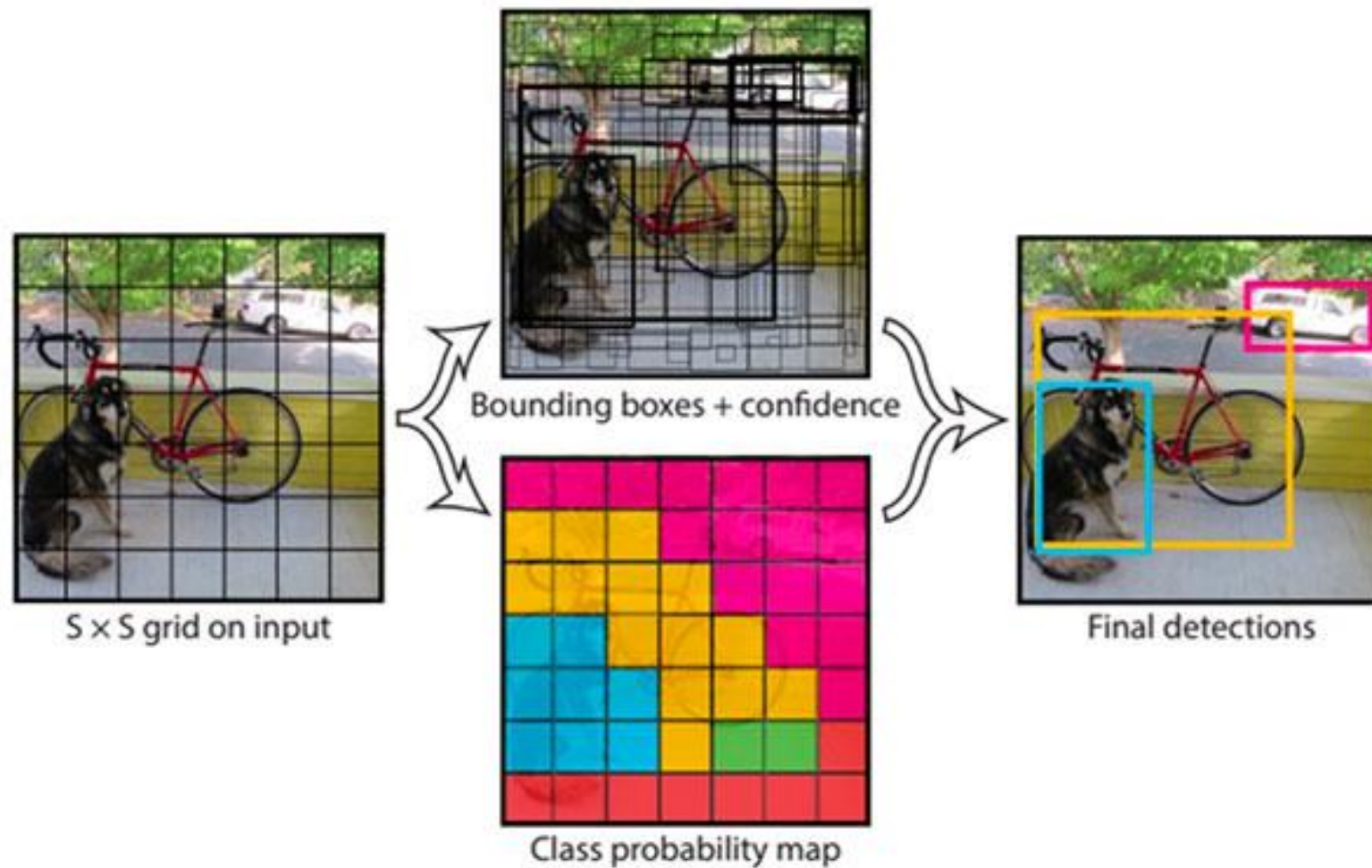


CAT, DOG, DUCK

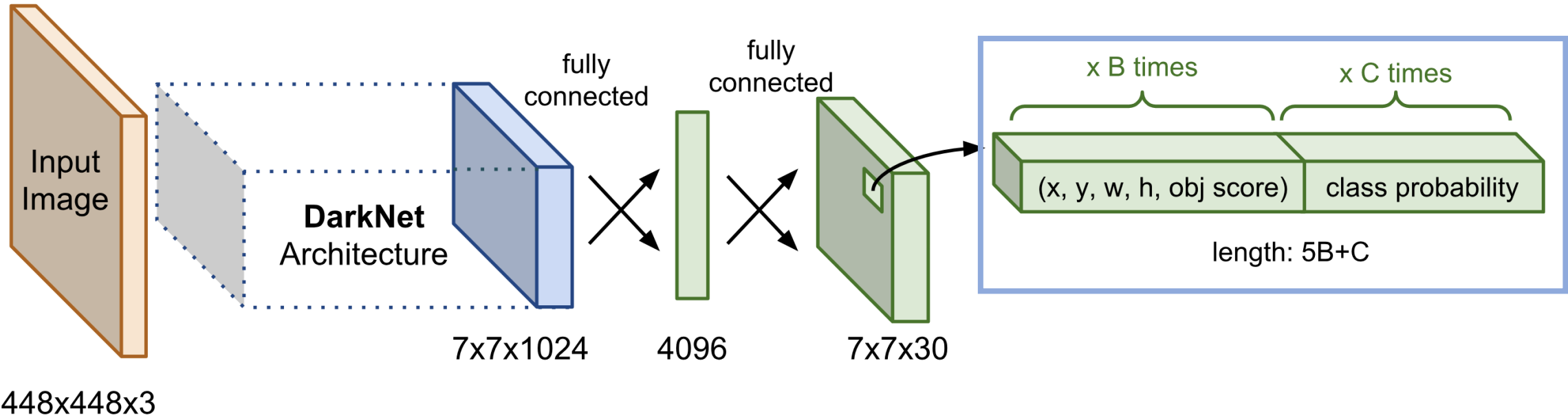


CAT, DOG, DUCK

Object Detection - Yolo

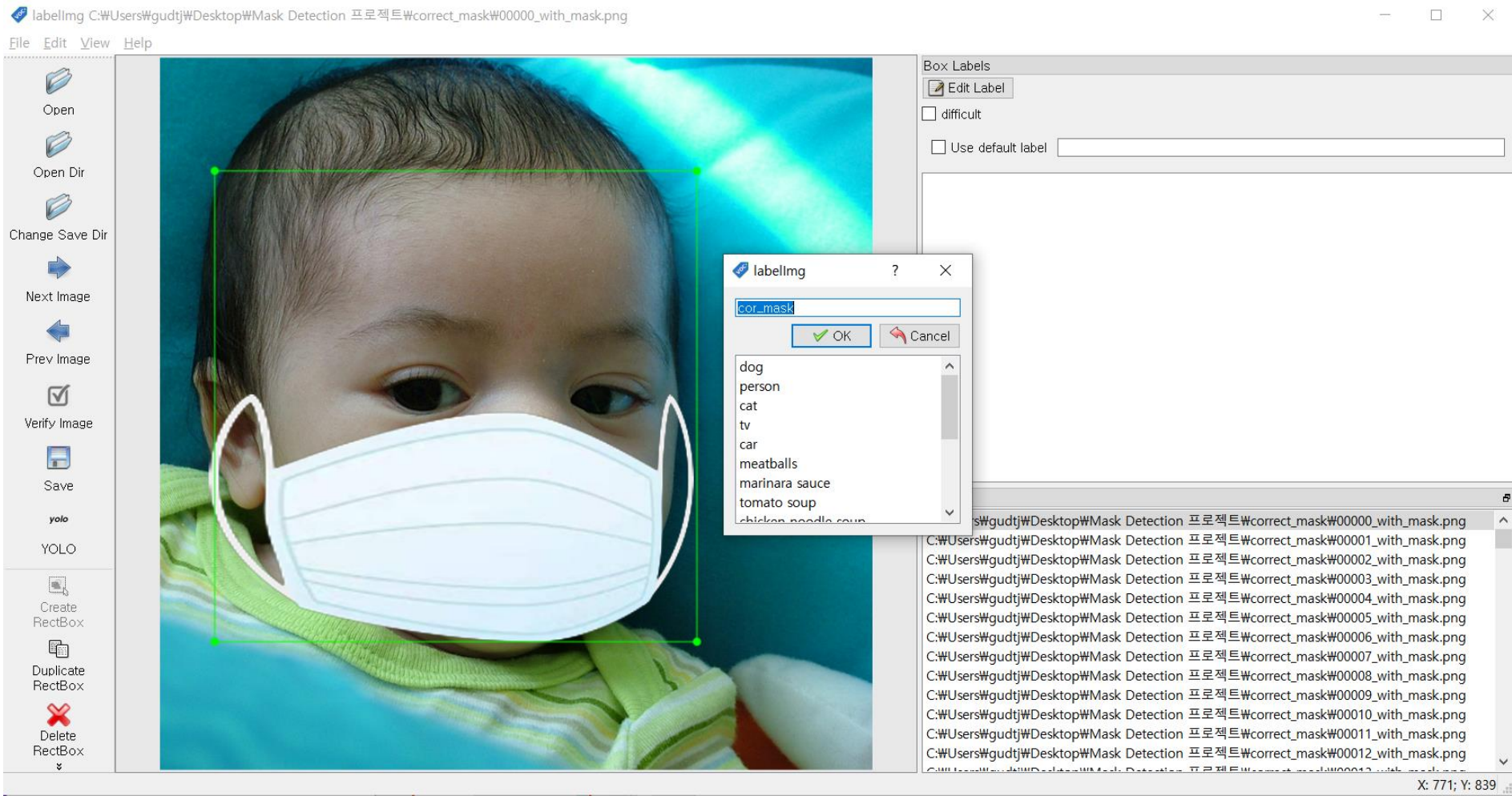


Object Detection - Yolo



Object Detection - Labelling

Labellmg

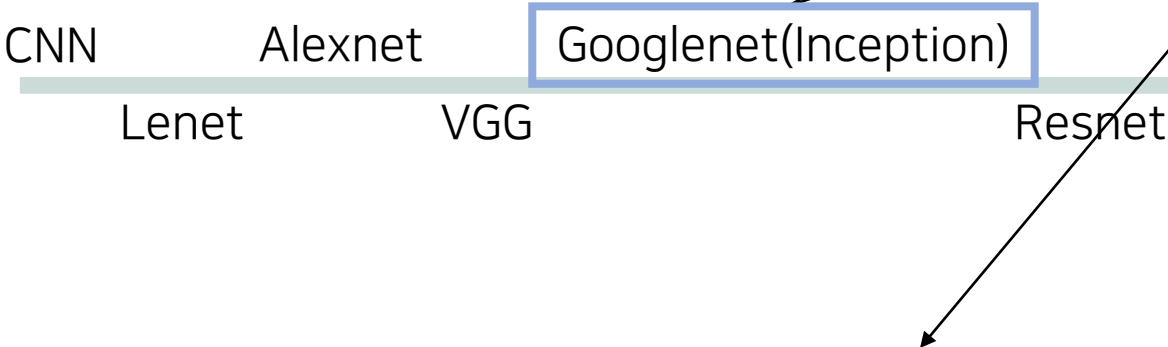


Class, Xmin, Xmax, Ymax, Ymin = [15 0.417969 0.483398 0.679688 0.667969]

04 Object Detection - Result

Input	Method	Accuracy
1500 images (correct, incorrect, no)	Pre-trained weight (Not-Freeze)	<pre>detections_count = 540, unique_truth_count = 301 class_id = 0, name = cor_mask, ap = 98.58% (TP = 96, FP = 6) class_id = 1, name = incor_mask, ap = 94.14% (TP = 94, FP = 24) class_id = 2, name = no_mask, ap = 96.78% (TP = 101, FP = 14) for conf_thresh = 0.25, precision = 0.87, recall = 0.97, F1-score = 0.92 for conf_thresh = 0.25, TP = 291, FP = 44, FN = 10, average IoU = 69.52 % IoU threshold = 50 %, used Area-Under-Curve for each unique Recall mean average precision (mAP@0.50) = 0.965018, or 96.50 % Total Detection Time: 13 Seconds</pre>

1) 수업시간에 배운 Image Classification model 에서 좀 더 발전된 모델들 학습 및 구현



2) 3가지 상황으로 나누어 Transfer Learning (pre-trained된 weight를 어떻게 사용해야 하는지, 어떤 층까지 Freeze를 시키고 학습시켜야 하는지, 모델의 어떤 부분을 수정해야 하는지 등)

3) Compressing the model using tensorflow

4) Uploading to the server

5) Labelling 진행 및 Object Detection model 구현

Image classification model을 backbone으로 가지고,
Region of proposal 과정이 추가되어 한 이미지 내의 다중 객체에 대해 인식하는
Object Detection model 구현

Image Classification model

SSD Yolo

Image Segmentation model

RCNN, Fast RCNN, Faster RCNN, EfficientNet

Mask_classifier.ipynb

: mobilenetv2, inceptionv3 훈련

Yolov3_implementation.ipynb

: yolov3 훈련

Try.py

: 비디오 혹은 실시간에서 시연

ApplicationDevelopment folder: 서버 관련 코드

Mask_Generator folder: 데이터셋 제작 관련 코드

Models folder: mobilenetv2, facet detector, yolov3 모델

감사합니다

Appendix

Object Detection - YOLOv3

