



COVID19

Mask Detection

최종발표

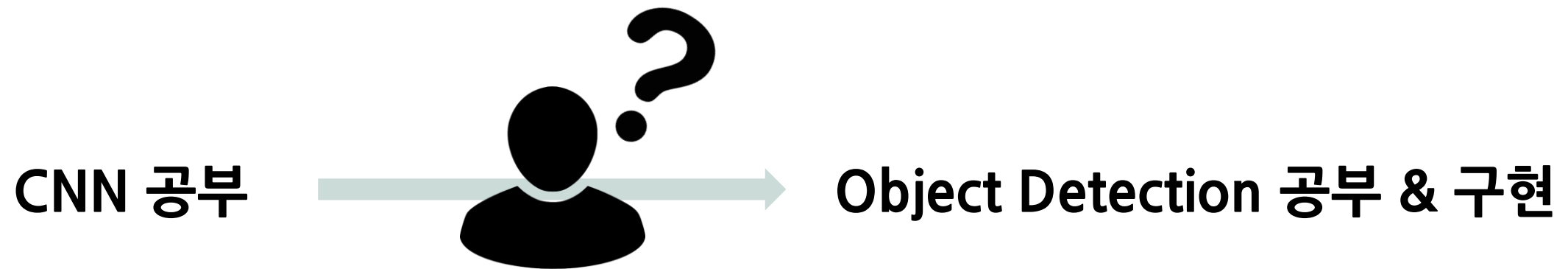
구형석 이노아
박기태 이지현

Goal of Project

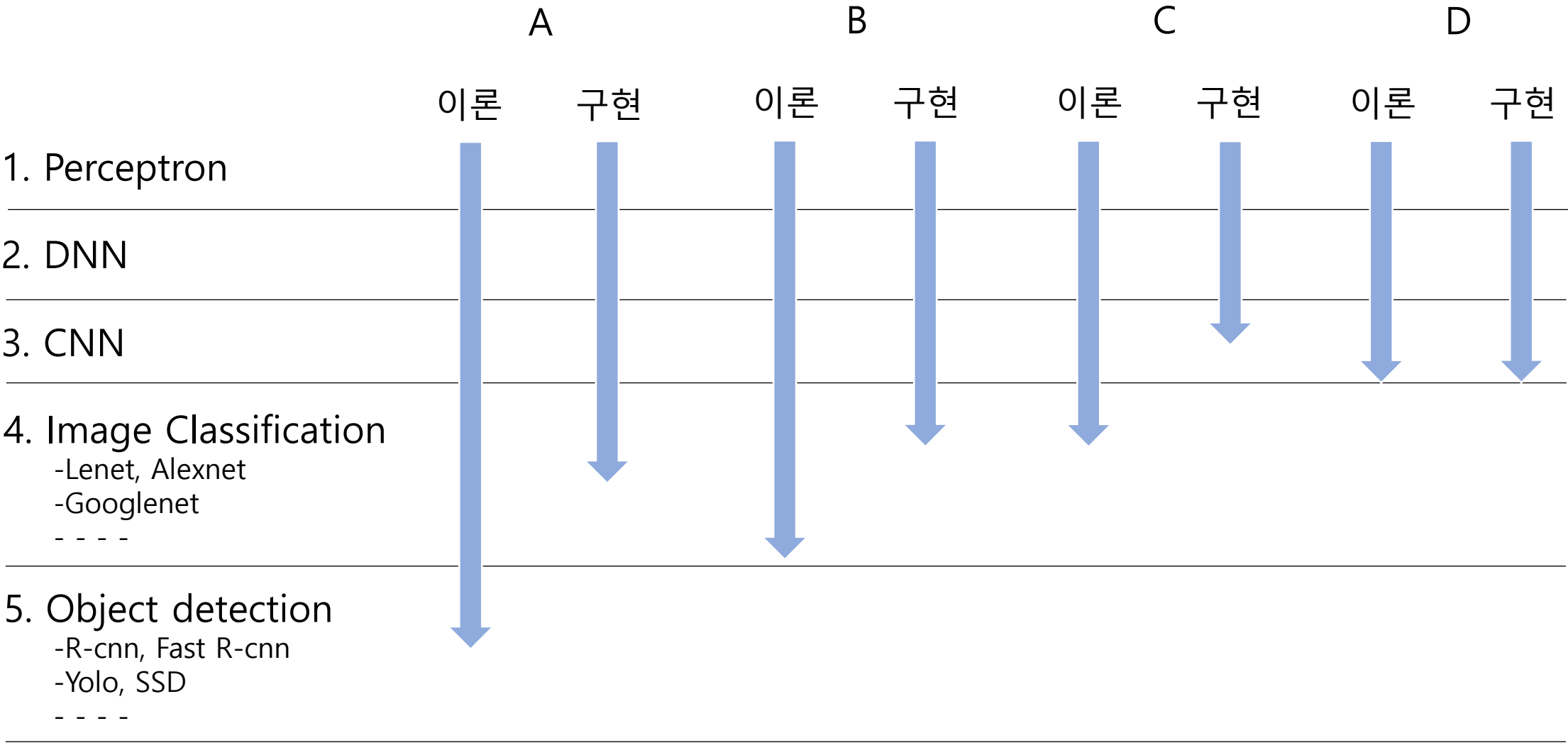
사진과 실시간 영상에서

Correctly, incorrectly, no mask를 분류해주는 Mask Detector 구현

00 Recap: 지난 학기 프로젝트?



프로젝트 시작 전 배경지식



주차별 진행 과정

1주차

프로젝트 관련 다양한 레퍼런스 탐색 및 역할분담 (~11/10)

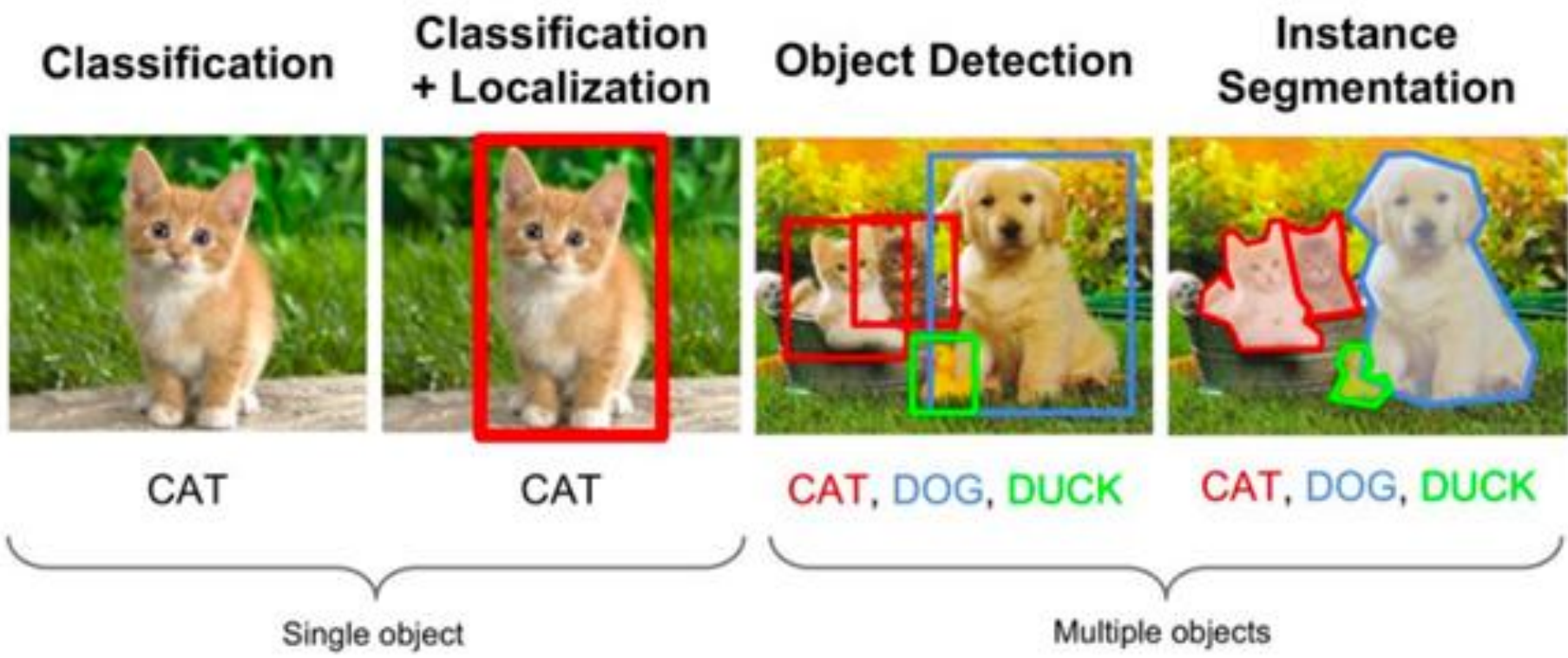
2주차

데이터 전처리, Mask classifier 구현
Lenet, Googlenet, Face Detection 스터디 (~11/19)

3주차

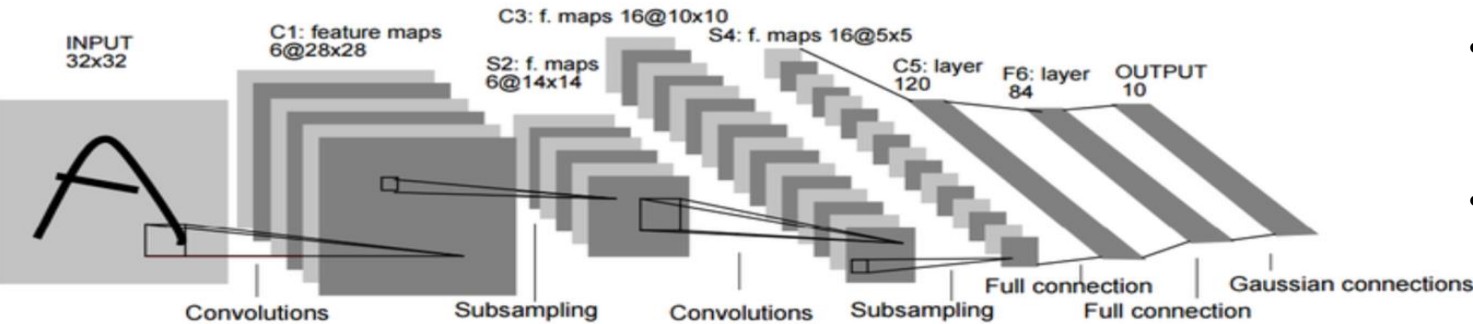
최종 모델 구현
ResNet, SSD, YOLO 스터디 (~11/24)

Background



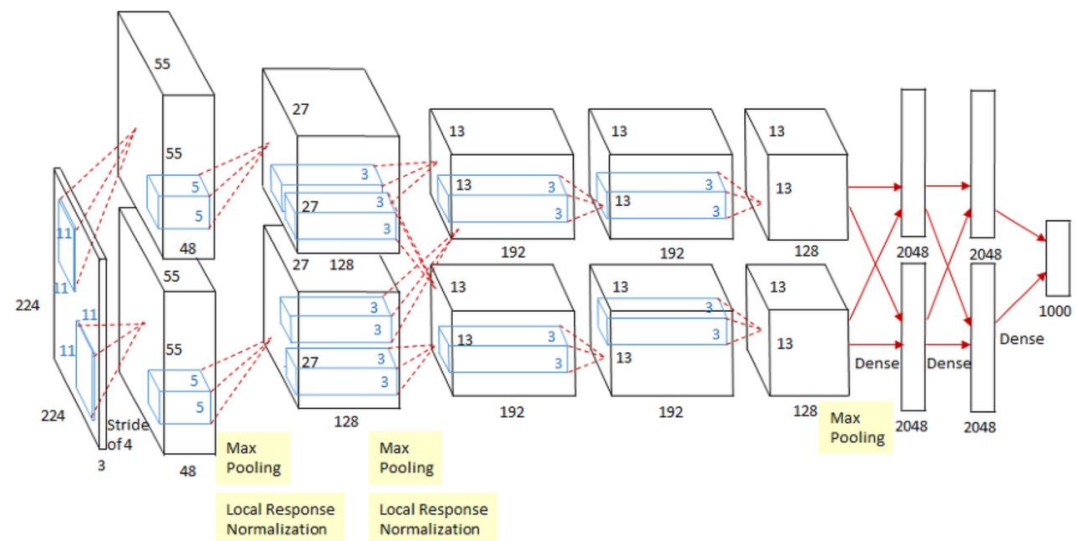
Background: Image Classification

[LeNet-5] 1998



- CNN 개념을 도입한 LeCun이 만든 CNN의 조상격 모델
- Convolution - Subsampling - Fc가 반복 되는 구조

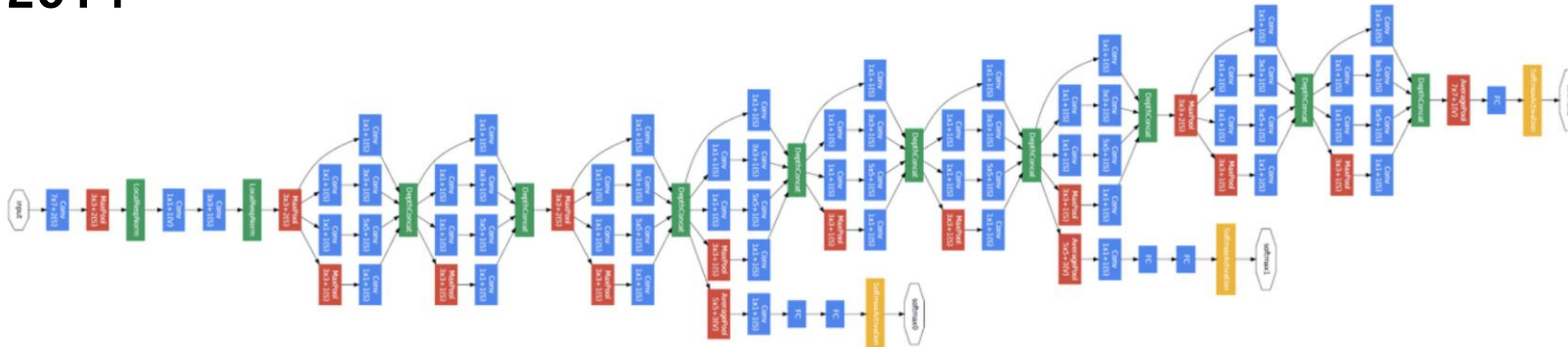
[AlexNet] 2012



- 2개의 GPU 연산을 위한 병렬적인 구조
- Relu , Drop out, Overlapping Max Pooling의 사용

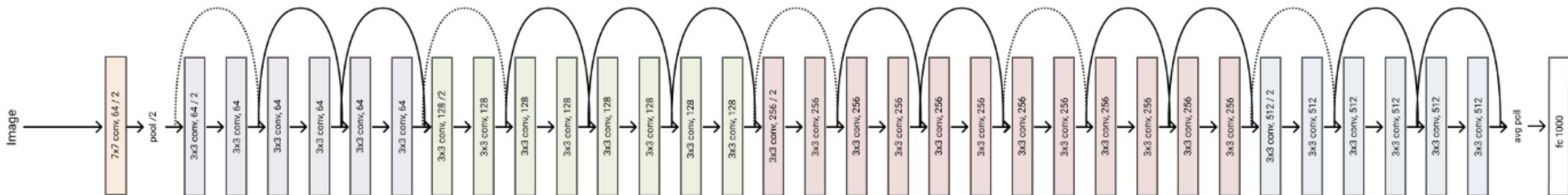
Background: Image Classification

[GoogLeNet] 2014



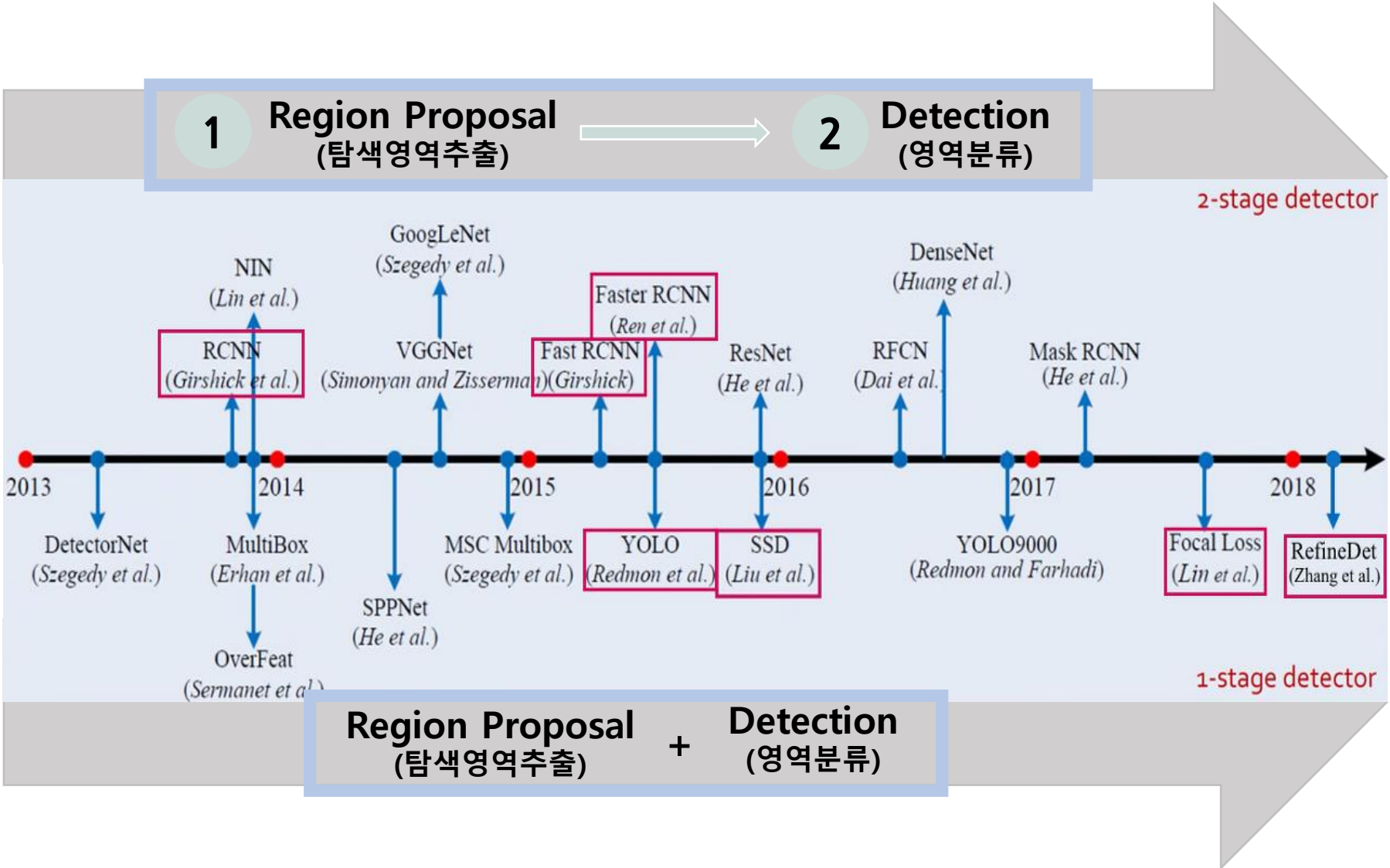
- Inception Module을 통해 여러 필터를 병렬적으로 사용하여 다양한 feature 추출
- 1X1 convolution과 bottleneck을 통해 연산량 감소

[ResNet] 2015



- Residual Block을 통해 모델이 학습하기 쉬운 구조로 만들어 줌
- 152개의 Layer로 이루어짐(ImageNet)

Background: Object Detection



모델 공부는 했는데 어떻게 구현할까?

1. 새로운 모델 개발, 고성능 모델에서의 성능 개선

- 석사, 박사, 현직 연구자들의 영역
- 최근 트렌드에서는 보통 많은 컴퓨팅 파워를 요구하는 것이 대부분

2. 프로젝트로 일상의 작은 문제에 적용해보기

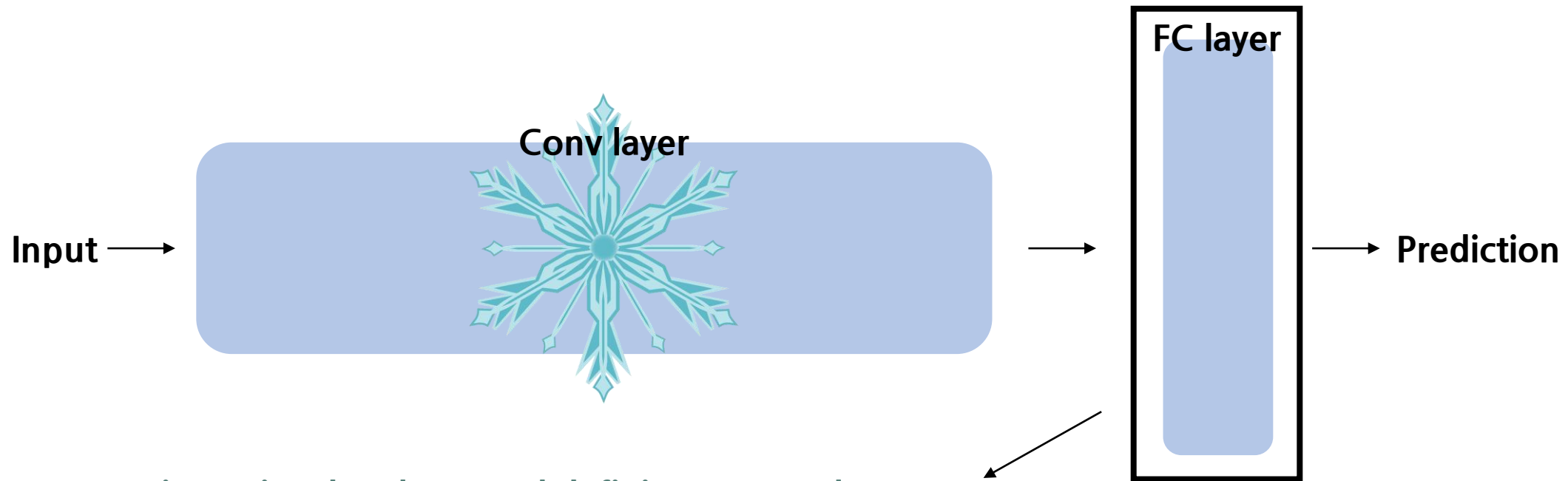
- 라이브러리에 내장된 모델 사용: Keras, opencv, dlib 등
- 오픈소스로 공개된 것에서 따오기: model 자체 혹은 코드
- Transfer learning 사용

01 Fine-tuning MobilenetV2

Transfer-Learning by using pre-trained MobilenetV2

Weight: Trained by "Imagenet"

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



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

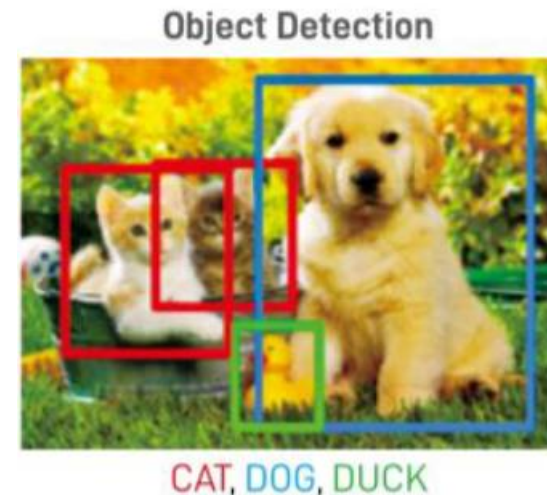
Fine-tuning MobilenetV2

```
1 # load the MobileNetV2 network, ensuring the head FC layer sets are left off
2 baseModel = MobileNetV2(weights="imagenet", include_top=False, input_tensor=Input(shape=(224, 224, 3)))
3
4 # construct the head of the model that will be placed on top of the the base model
5 headModel = baseModel.output
6
7 headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
8 headModel = Flatten(name="flatten")(headModel)
9 headModel = Dense(128, activation="relu")(headModel)
10 headModel = Dropout(0.5)(headModel)
11 headModel = Dense(3, activation="softmax")(headModel)
12
13 # place the head FC model on top of the base model (this will become the actual model we will train)
14 model = Model(inputs=baseModel.input, outputs=headModel)
15
16 # loop over all layers in the base model and freeze them so they will *not* be updated during the first training process
17 for layer in baseModel.layers:
18     layer.trainable = False
```

Why Classification model?

Why not using Object Detection model?

- 1) Box Label 데이터의 확보에 대한 부담
- 2) Object Detection model을 직접 transfer learning 했을 때,
pre-trained face-detector보다 월등한 성능을 낼지 불확실



Instead, using image classifier combining with the face detector

MobileNet: Fast and light model with less information loss and memory usage

Classification model + Face detector

Classification model - 하나의 사진에 나와 있는 여러 명을 인식할 수 없음

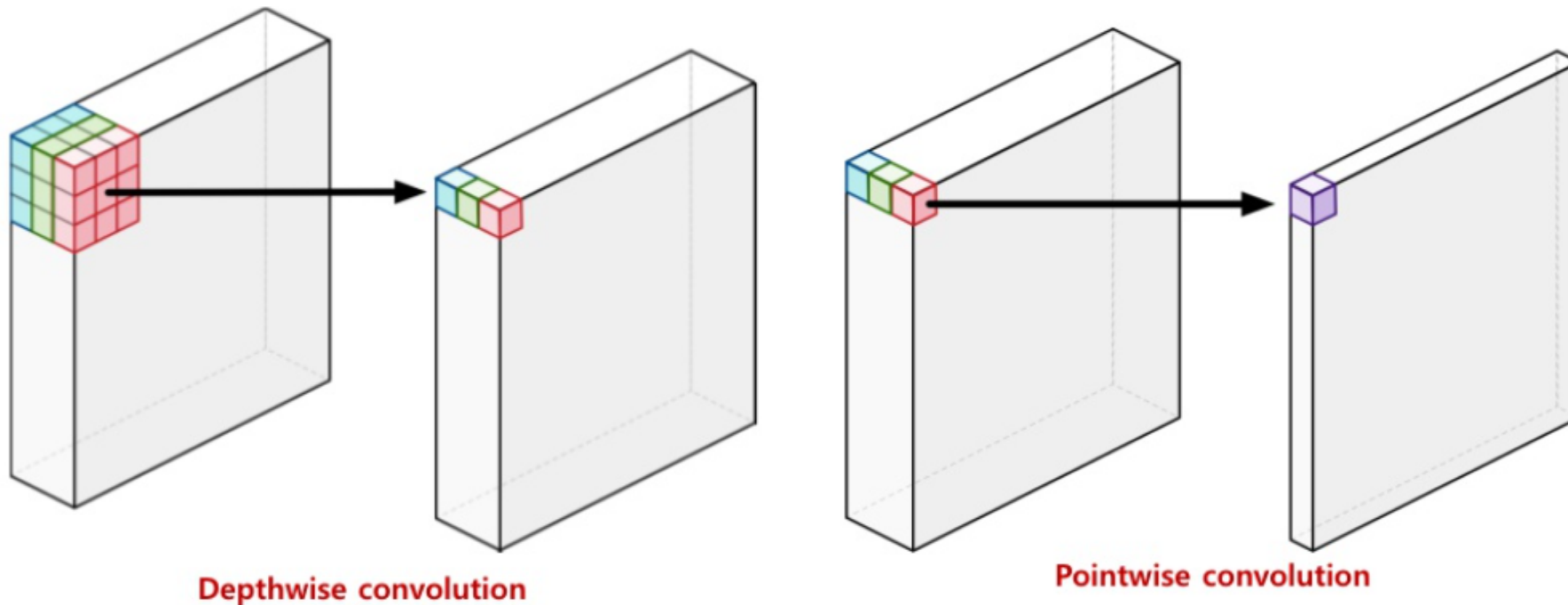


사진에서 Face detector로 얼굴 부분을 추출

얼굴 부분을 classifier에 넘겨 mask를 썼는지 안 썼는지 판단!

Depthwise Separable convolution

- Depthwise Convolution + Pointwise Convolution(1x1 convolution)

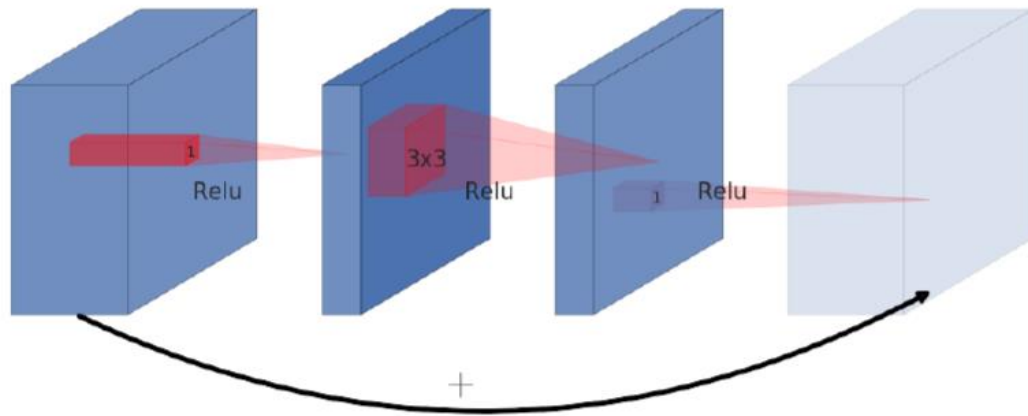


Figures from <http://machinethink.net/blog/googles-mobile-net-architecture-on-iphone/>

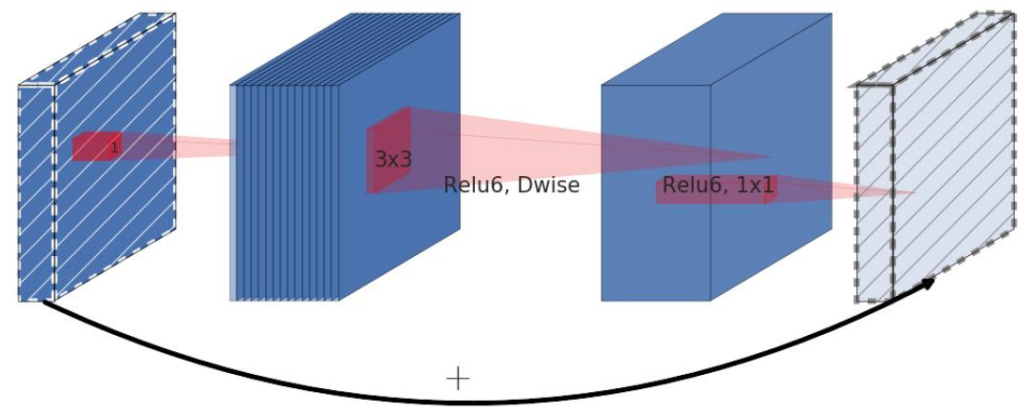
01 Understanding MobileNetV2

Inverted residual

Original residual block



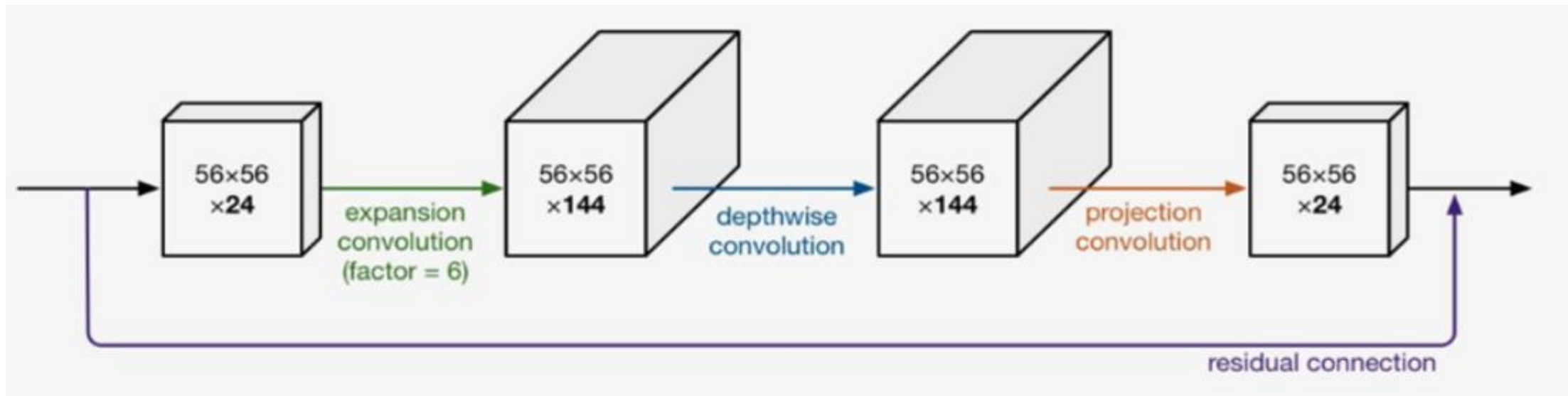
Inverted residual



Because of bottleneck, it loses little information even the residual has narrow channel

01 Understanding MobileNetV2

Build fast and light model
with less information loss and memory usage

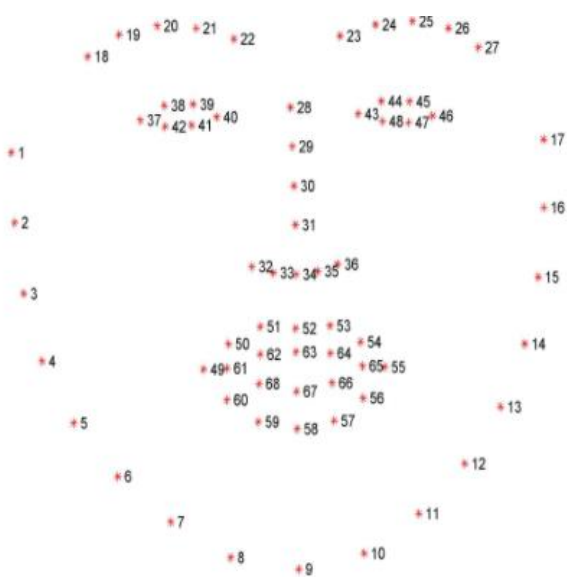


Dataset Preparation: Generation Process

Face Location와 Face Landmark(68개 index) 찾기

코와 턱 찾아 마스크를 올려놓고 사이즈 조절하기

마스크 회전 및 얼굴의 중앙에 위치시키기



왼쪽과 오른쪽의 비율을 계산하여
왼쪽마스크, 오른쪽마스크 두 부분으로 나누어 최종적으로 합치는 과정 수행

02 Dataset Preparation: Results



0.jpg



1.jpg



2.jpg



3.jpg



4.jpg



5.jpg



6.jpg



7.jpg



8.jpg



5_with_mask.jpg



6_with_mask.jpg



7_with_mask.jpg



8_with_mask.jpg



0_with_mask.jpg



1_with_mask.jpg



2_with_mask.jpg

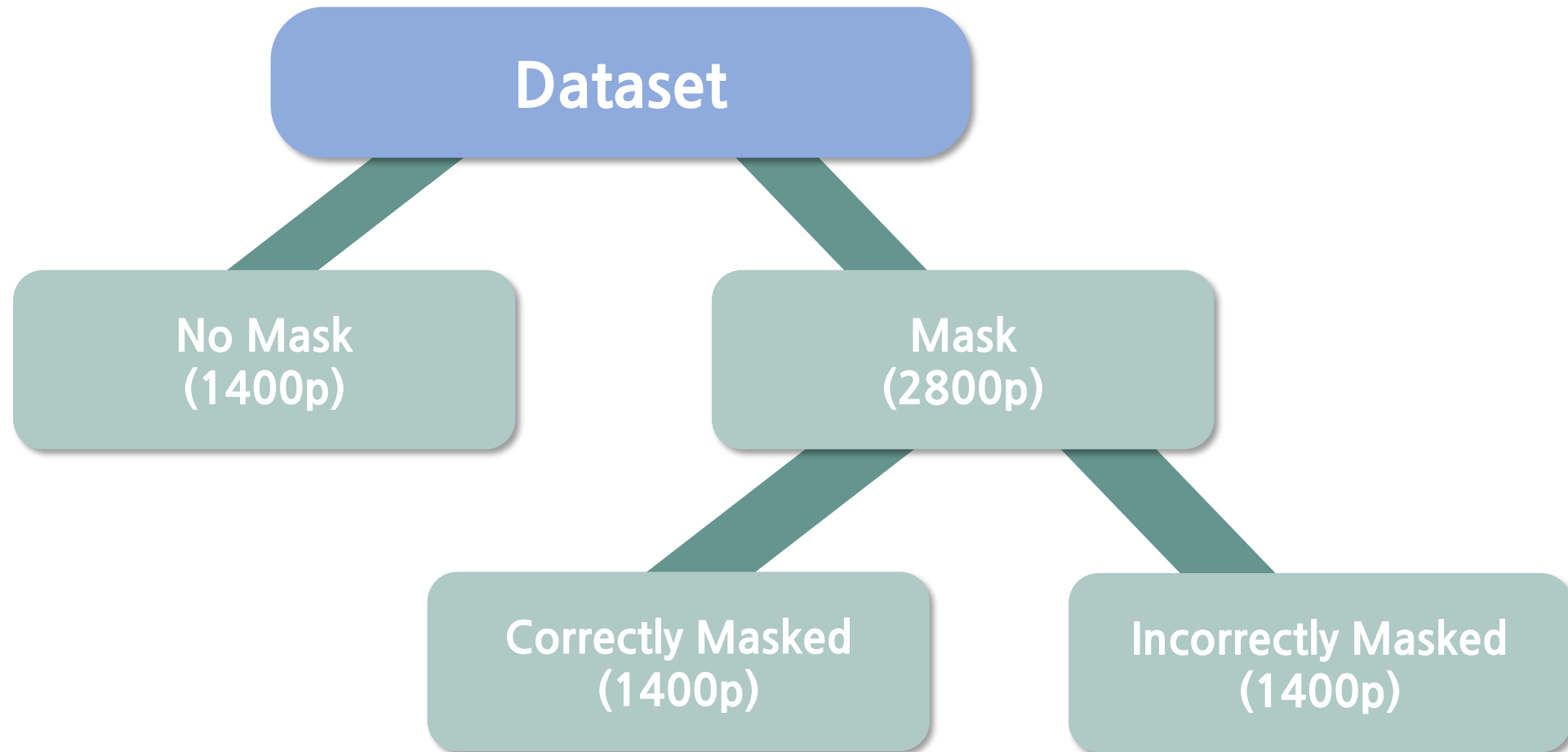


3_with_mask.jpg



4_with_mask.jpg

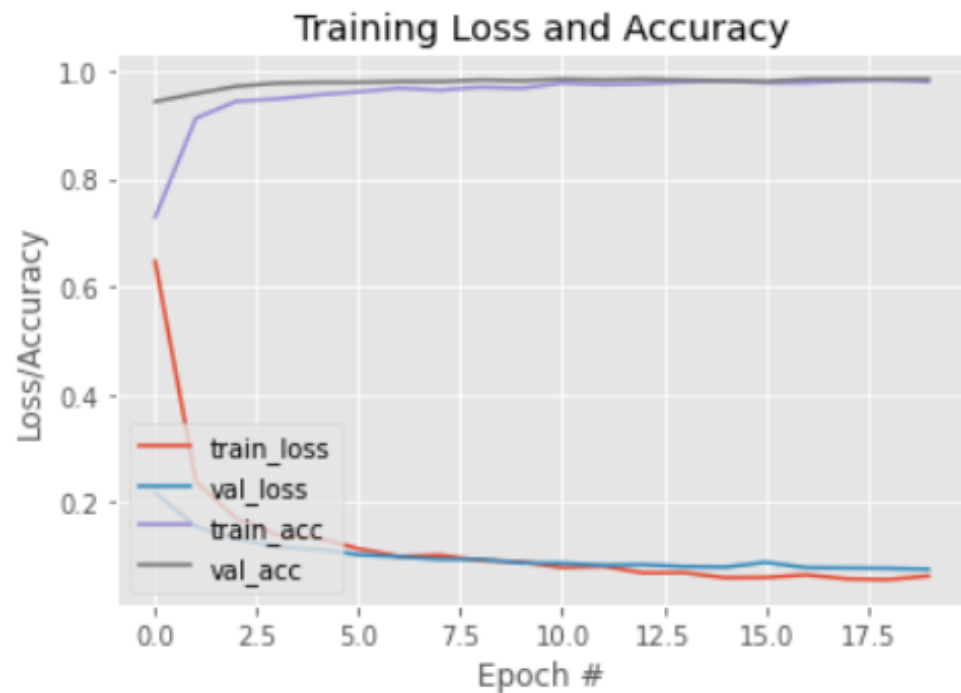
02 Dataset Preparation: Results



02 Preprocessing: MobileNetV2

- 01 Input Image
- 02 Resizing to 224x224
- 03 Image to array
- 04 Scaling pixel intensity $\rightarrow [-1,1]$

Accuracy of the mask classifier



Accuracy of test set

	precision	recall	f1-score	support
cor_mask	0.99	0.98	0.99	280
incor_mask	0.98	0.99	0.99	280
nomask	0.99	1.00	0.99	280
accuracy			0.99	840
macro avg	0.99	0.99	0.99	840
weighted avg	0.99	0.99	0.99	840

Face detector

Face detection?

Haar Cascade 방법: 얼굴에서 공통적으로 두드러지는 명암의 대비를 통해 얼굴 탐색

DNN 방법: Deep Neural Net을 활용해 얼굴 탐색, 기존 Cascade classifier보다 더 좋은 성능

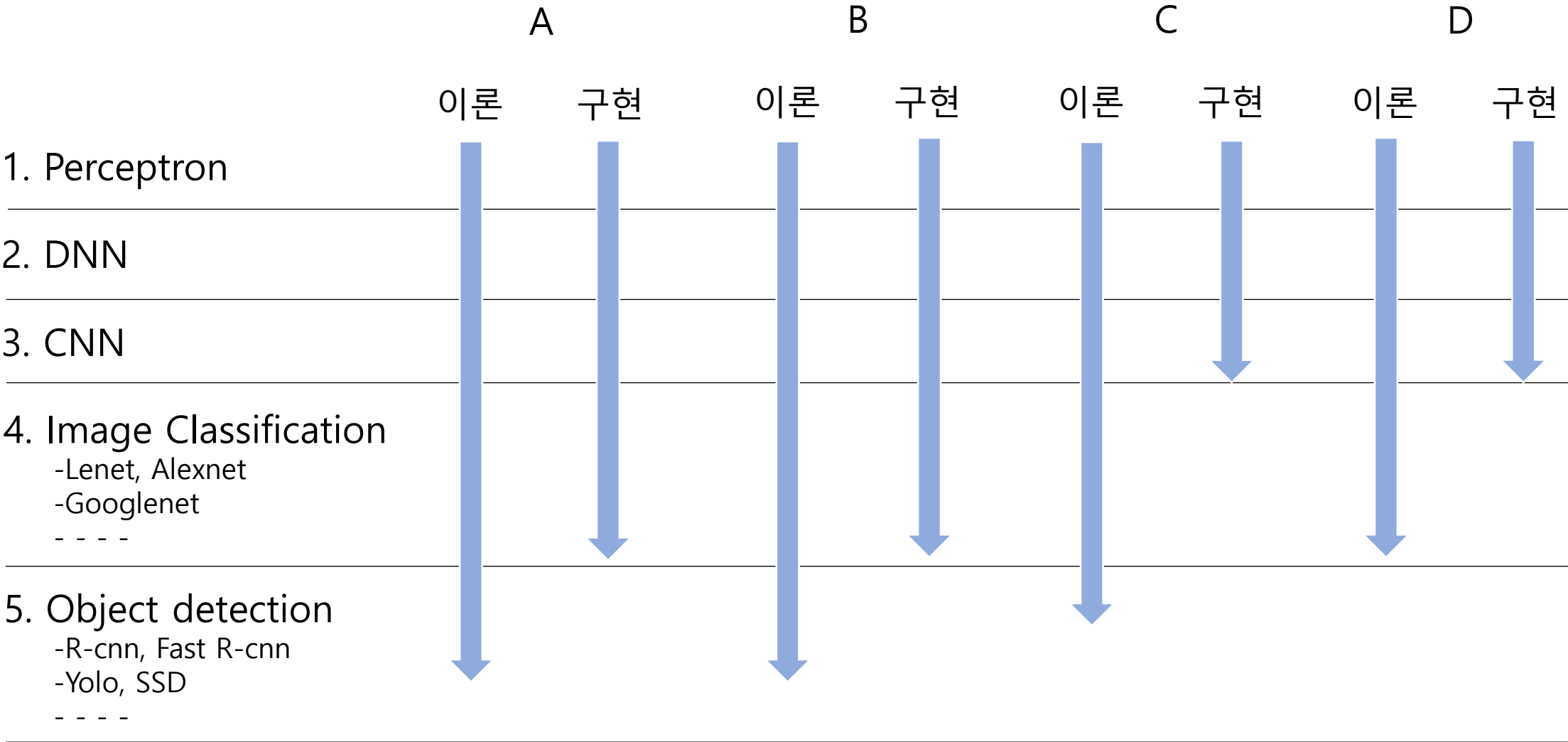
Opencv 제공 Face Detector

: SSD와 ResNet-10 기반, caffe 프레임워크로 구현

: blob(binary large object, 멀티미디어 자료를 담을 수 있는 형태)을 input으로 받음



시연 영상



감사합니다