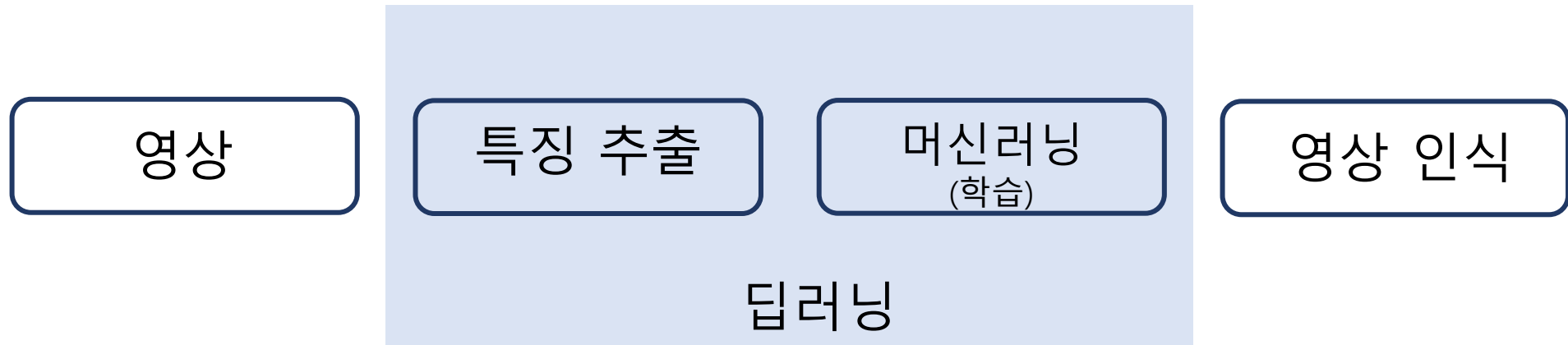


# Computer Vision

OpenCV와 딥러닝을 활용하는  
간단한 방법과 예제

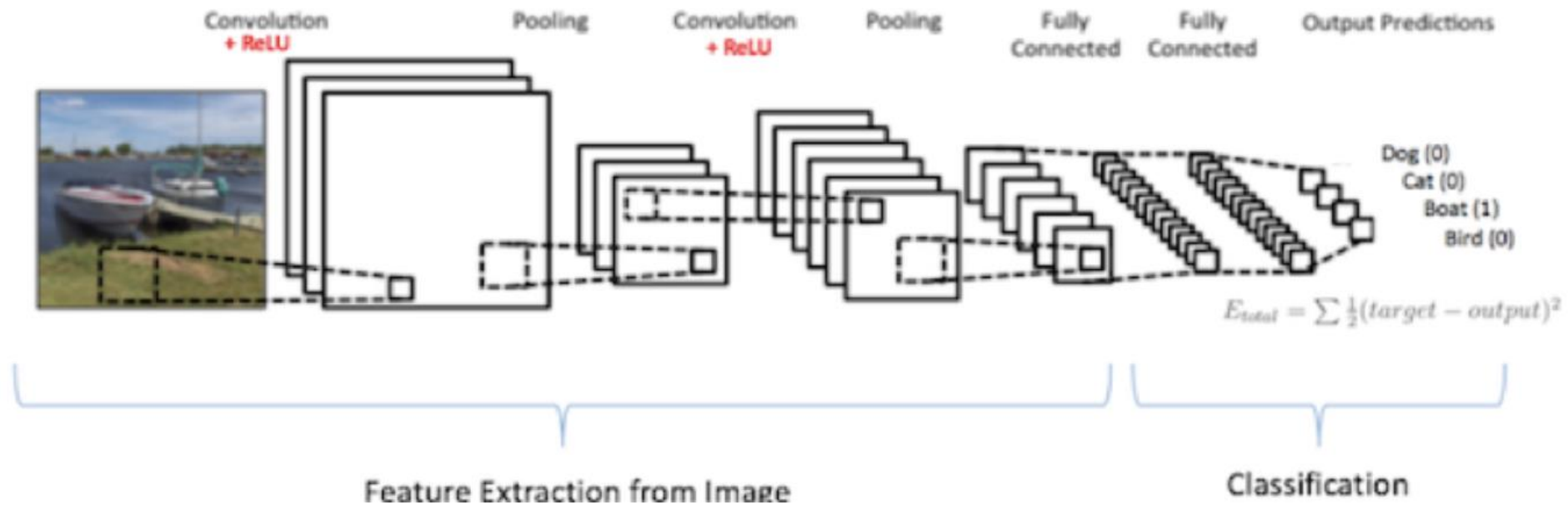
소프트웨어학과 김민지

# 머신러닝과 딥러닝 학습 과정



# 합성곱신경망 (CNN)

CNN 세가지 종류 층 ( 컨볼루션층 / 풀링층 / 완전연결레이어 )



# OpenCV 4.5.0 is here!

The best computer vision library in the world got even better.

## OpenCV : 딥러닝 모듈(Dep Neaural Network) 제공

OpenCV는 이미지를 처리하는 유용한 라이브러리에서  
딥러닝 모듈이 추가 되면서 컴퓨터 비전 분야의 활용력이 높아짐

# OpenCV Tutorials

---

- **Introduction to OpenCV** - build and install OpenCV on your computer
- **The Core Functionality (core module)** - basic building blocks of the library
- **Image Processing (imgproc module)** - image processing functions
- **High Level GUI and Media (highgui module)** - built-in graphical user interface
- **Image Input and Output (imgcodecs module)** - read and write images from/to files using *imgcodecs* module
- **Video Input and Output (videoio module)** - read and write videos using *videoio* module
- **Camera calibration and 3D reconstruction (calib3d module)** - extract 3D world information from 2D images
- **2D Features framework (feature2d module)** - feature detectors, descriptors and matching framework
- **Video analysis (video module)** - algorithms for video streams: motion detection, object and feature tracking, etc.
- **Object Detection (objdetect module)** - detect objects using conventional CV methods
- **Deep Neural Networks (dnn module)** - infer neural networks using built-in *dnn* module
- **Machine Learning (ml module)** - machine learning algorithms for statistical classification, regression and data clustering
- **Graph API (gapi module)** - graph-based approach to computer vision algorithms building
- **Computational photography (photo module)** - advanced photo processing
- **Images stitching (stitching module)** - create panoramas and more using *stitching* module
- **OpenCV iOS** - running OpenCV on an iDevice
- **GPU-Accelerated Computer Vision (cuda module)** - utilizing power of video card to run CV algorithms

# Detection

- **Object Detection**
- **Cascade Classifier**
- [https://docs.opencv.org/master/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/master/db/d28/tutorial_cascade_classifier.html)
- **Deep Neural Networks (dnn module)**
- [https://docs.opencv.org/master/d2/d58/tutorial\\_table\\_of\\_content\\_dnn.html](https://docs.opencv.org/master/d2/d58/tutorial_table_of_content_dnn.html)

# Haar Object detection

Rapid Object Detection

Haar Cascade 를 이용한 얼굴 인식

눈 인식 = harrcascade\_eye\_tree\_eyeglasses.xml

얼굴 인식 = haarcascade\_frontalface\_alt.xml

Code (son2.video + tedy.video)

# Dnn module

Code (son2.video + tedy.video)





## YOLO: Real-Time Object Detection

---

You only look once (YOLO) is a state-of-the-art, real-time object detection system. On a Pascal Titan X it processes images at 30 FPS and has a mAP of 57.9% on COCO test-dev.

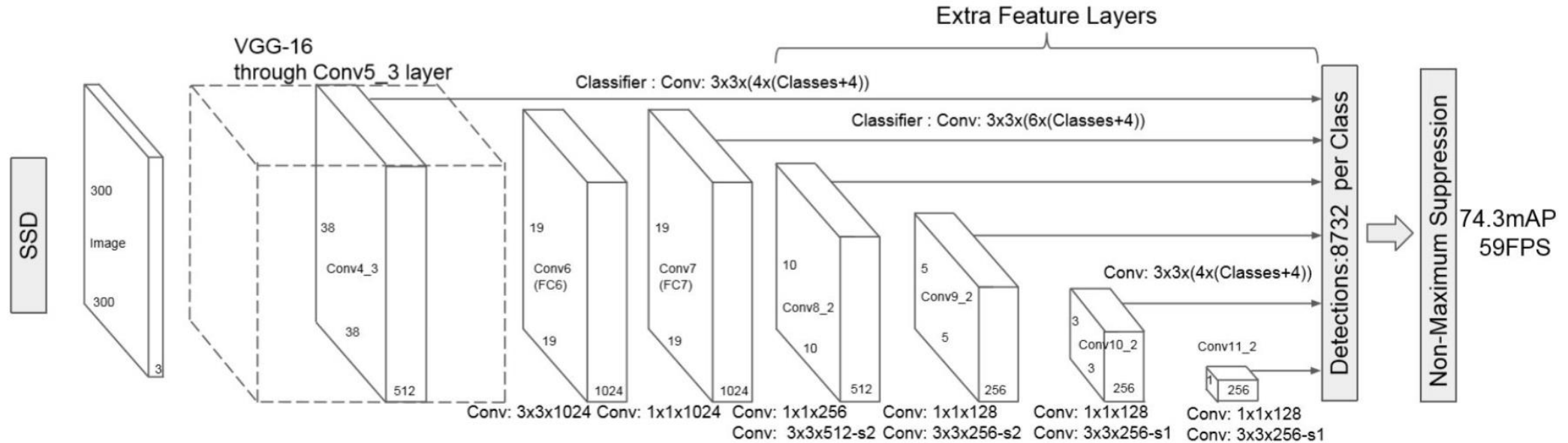
**yolo = you only look once**

모델 다운로드:

<https://pjreddie.com/darknet/yolo/>

```
classes = ["person", "bicycle", "car", "motorcycle", "airplane", "bus", "train",  
"truck", "boat", "traffic light", "fire hydrant", "stop sign", "parking meter",  
"bench", "bird", "cat", "dog", "horse", "sheep", "cow", "elephant", "bear",  
"zebra", "giraffe", "backpack", "umbrella", "handbag", "tie", "suitcase",  
"frisbee", "skis", "snowboard", "sports ball", "kite", "baseball bat", "baseball  
glove", "skateboard", "surfboard", "tennis racket", "bottle", "wine glass",  
"cup", "fork", "knife", "spoon", "bowl", "banana", "apple", "sandwich",  
"orange", "broccoli", "carrot", "hot dog", "pizza", "donut", "cake", "chair",  
"couch", "potted plant", "bed", "dining table", "toilet", "tv", "laptop",  
"mouse", "remote", "keyboard", "cell phone", "microwave", "oven", "toaster",  
"sink", "refrigerator", "book", "clock", "vase", "scissors", "teddy bear", "hair  
drier", "toothbrush" ]
```

# The Single Shot Detector(SSD)



*SSD discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location (multiple feature map).*

# Face\_recognition

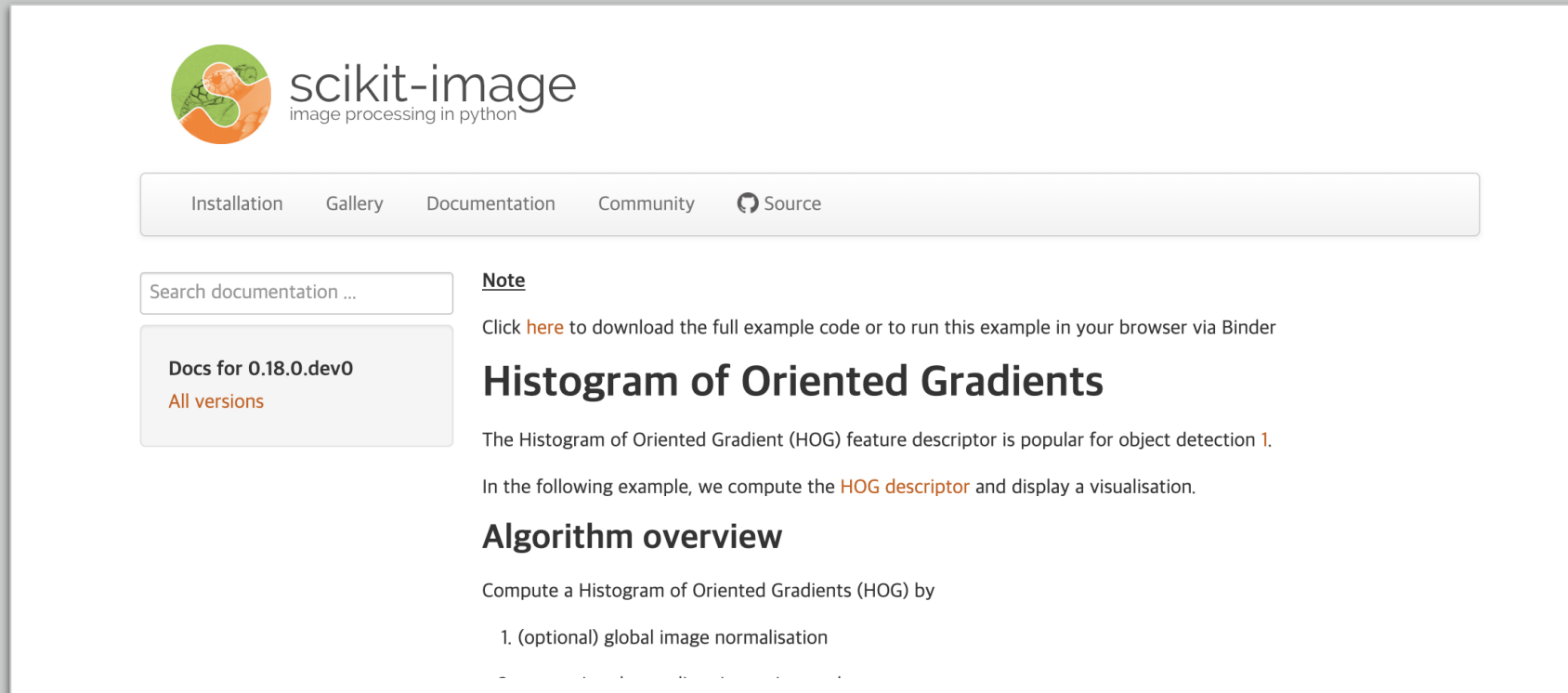
사람의 얼굴 특성을 학습시켜서 사람 인식  
= Face\_recognition 과 Dlib 모듈 설치

Encoding 작업 -> cnn

Face\_recognition -> cnn / hog

# Face\_recognition(Hog)

The Histogram of Oriented Gradient (HOG)  
좀 더 윤곽을 인식하기 쉽게 변환해주는 것



The screenshot shows the scikit-image documentation page for the Histogram of Oriented Gradients (HOG) feature descriptor. The page has a header with the scikit-image logo and navigation links: Installation, Gallery, Documentation, Community, and Source. Below the header is a search bar and a sidebar with links to documentation for version 0.18.0.dev0 and all versions. The main content area includes a note about downloading example code or running it via Binder, followed by the title "Histogram of Oriented Gradients". The text explains that HOG is popular for object detection and provides an example of computing the HOG descriptor and displaying a visualization. The "Algorithm overview" section begins with the instruction to compute a Histogram of Oriented Gradients (HOG) by:

1. (optional) global image normalisation