OBJECT DETECTION APPLICATION

# Overview

There are 2 ways to build an object detection application:

* (1) Using a deep learning pretrained model by others
* (2) Train the deep learning model by our own way

There are pros and cons for each method as well. For example,

* If we select (1), then we depend on the author of the pretrained model which we are using. It means that the performance of our application depends on the current pretrained model we are using.
* If we select (2), then we build and train models from scratch. We have our own abilities to improve the performance of the deep learning model. However, training the deep learning model also costs time and big resources.

In the next sections, I will discuss about approach #2. If we want to go on with approach #1, just remove a few components from the system.

# Architecture

There will be 3 important part of the application system:

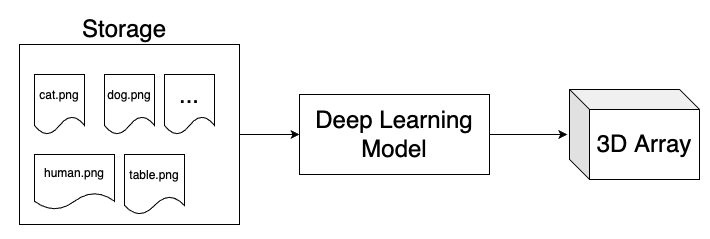
* Storage
* Deep Learning Model
* Frontend
* Backend

## **Storage**:

It is the place we place training images with their labels. It can be AWS S3 or gcloud storage. As we are aware, an image is also a 3D array of float numbers (width x height x channel) where channel = 3 representing RGB colors. So, we can store each image ‘s 3D array in SQL or NoSQL database, of course. However, the cost of DB is much more expensive than the cost of storage like AWS S3 / gcloud storage. That ‘s why I suggested using storage for storing images.

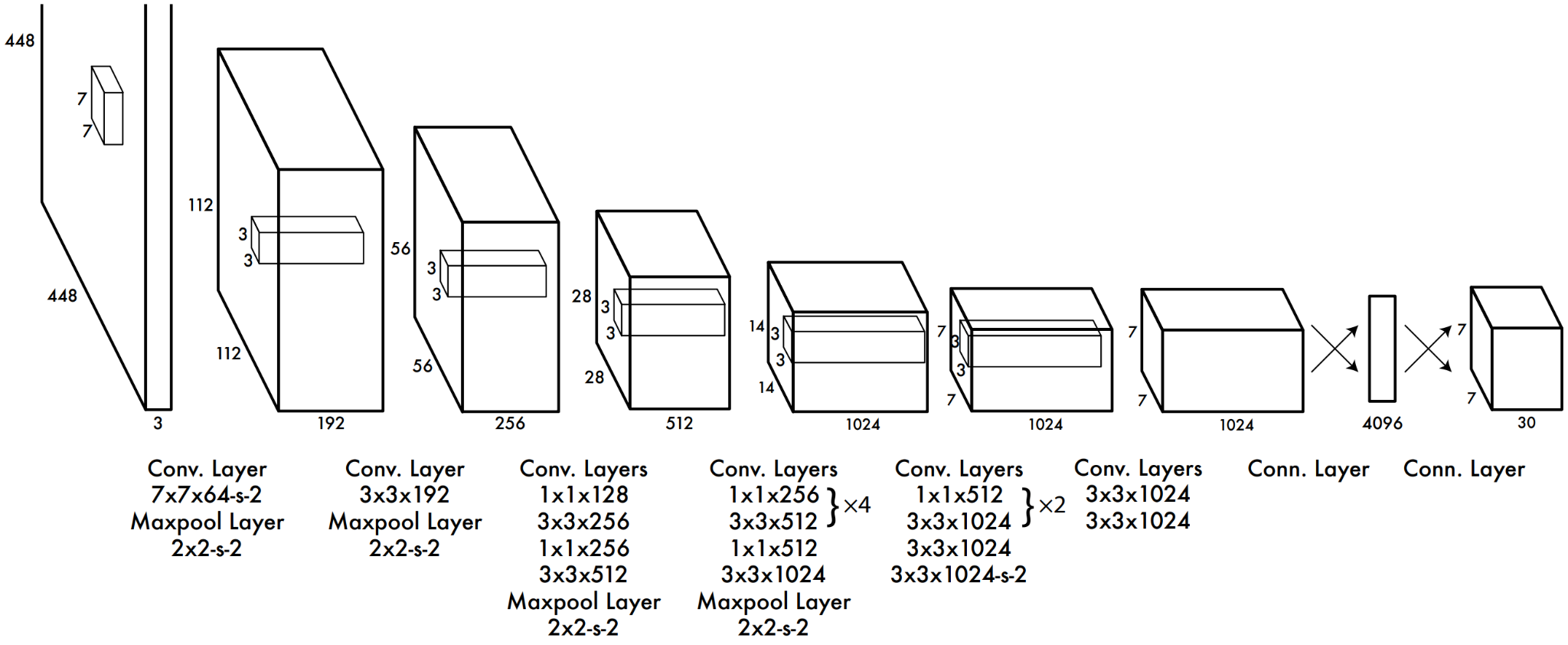
## **Deep Learning Model:**

It is the component to get all the training images & their labels (for example cat / dog / human / sofa / etc) from the storage and run the deep learning training model. This component requires a very **high performance computing** instance for training millions of images. For example, we can use AWS EC2 or gcloud VM with 8 - core CPU and 64G RAM. Of course, it depends on the number of training images as well as the maximum required time (hours) for training step, how strong or big the CPU and RAM is.



In order to estimate the RAM, we need to understand the architecture of the deep learning model. For example, if we use YOLO network, then we know that the input image for the YOLO network might be in the shape of 608x608x3. Then, we want to batch 100 images for each training step. Then, for the batch of 100 images, we need 100 x 608 x 608 x 3 x 8 bytes (1 float = 8 bytes) = 846MB. It is only the input image batch.

Looking at the YOLO architecture, we can see that the input images will go through some layers and each layer is also a 3D or 2D array. We can compute how much memory we need for each layer as well. Then, we will know exactly how much RAM we need for this deep learning model.



## **Frontend:**

The user will interact with the application via UI provided by the Frontend component. We don’t need a very high instance for this part. In general, if we use Angular framework for coding the frontend, we can deploy it onto firebase and it is free.

## **Load balancer:**

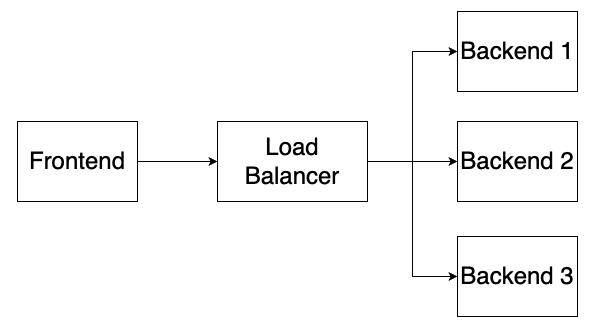
The request from frontend will go to the load balancer first. It is one way for us to scale the system horizontally (Vertical scale means that we will increase the backend server).

The work balancer will forward or distribute the requests from frontend to a specific backend server based on workload of each backend.

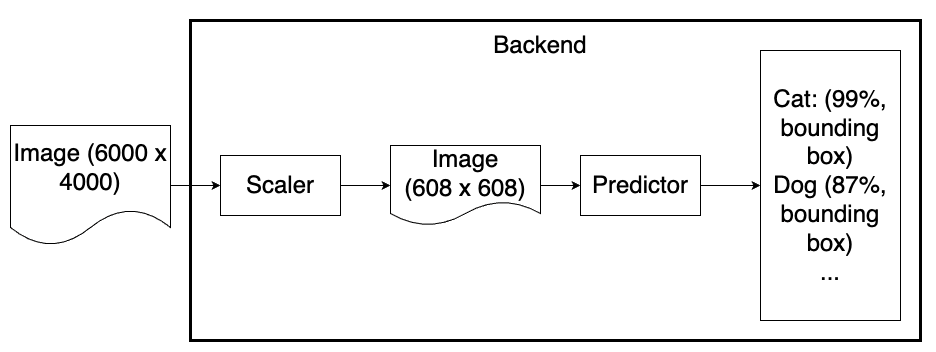
## **Backend:**

Based on the traffic of frontend requests, we will decide:

* How many backends we need for this application
* How strong the backend instance will be



The detailed architecture of the backend is explained as per below picture. Remember from the deep learning model component, its output is a 3D Array pretrained param. Then, the 3D Array pretrained params will be used by the predictor of the backend to predict whether there are cats / dogs / human / etc in the picture or not.



The RAM and CPU needed for the backend is not required to be high performance at all. If there is only 1 request / min from frontend, as per my experience, we only need a normal instance with 2-core CPU and 3.75G RAM (of course, we can use a swap memory of 16G RAM for provision if needed).