1. Questions in Computer Vision / AI

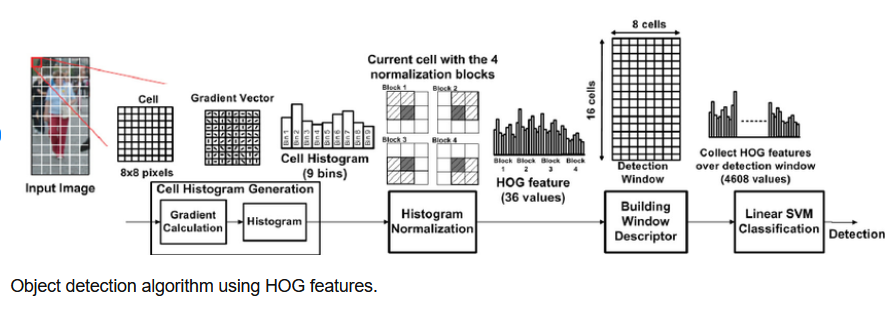
1. Discuss some Object Detection methods used in the Computer Vision field.

Answer:

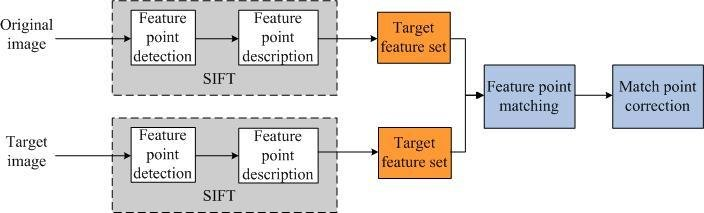
Generally, there are two different approaches of object detection methods

a/ Non-neural approach:

* HOG: counts occurrences of gradient orientation in localized portions of an image (patch) through analyzing the histogram as the below figure. A patch may come from an object, a person, meaningless background, or anything else, and is merely a way to describe an area using edge information. The HOG descriptor has a few key advantages over other descriptors. Since it operates on local cells, it is invariant to geometric and photometric transformations, except for object orientation. That is the reason why this method give a good performance for pedestrian detection with a static camera.



* SIFT is a computer vision algorithm to detect, describe, and match localfeatures in images. It is rotation and scale invariant, but mathematically complicated and computationally heavy.



b/ Neural network approach:

* R-CNN: e.g. Pretrained ResNet-101 architecture used for vehicle extraction. The first subnetwork is a region proposal network trained to classify objects from the background, and the second subnetwork is trained to classify the detected objects alone (car, pedestrian, etc.). There are many different advanced models of RCNN:
  + Fast RCNN
  + Faster RCNN
  + Mask RCNN
* Single Shot Detection (architectures such as RetinaNet, YOLOv3, etc.)
* YoloV5 is one of the fastest and most accurate models used for real-time object detection

2. Discuss some approaches to do model optimization in the Deep Learning area.

The area of model optimization can involve various techniques with two main targets:

a/ An acceptable model selection and tuning: hyper-parameter (weights/ thresholds/ window sizes/ transformation) tuning

* Standard tuning methods: manual search, random search, grid search
* Optimization feedback loop: sigopt tool
* Bayesian global optimization is expensive while training on the big data

b/ Model optimization for higher accuracy and real-time inference for resource-constrained devices:

* Reduce parameter count with pruning and structured pruning. Pruning is one model compression technique that allows the model to be optimized for real-time inference for resource-constrained devices. It was shown that large-sparse models often outperform small-dense models across various different architectures
* Reduce representational precision with quantization.
* Update the original model topology to a more efficient one with reduced parameters or faster execution

3. Discuss some standard feature extraction models with focus on Deep Learning.

### **AlexNet**

AlexNet is a convolutional neural network composing of 8 layers, where 5 are convolutional and 3 are fully connected.. AlexNet is used in object detection models such as R-CNN and HyperNet.

### **VGG-16**

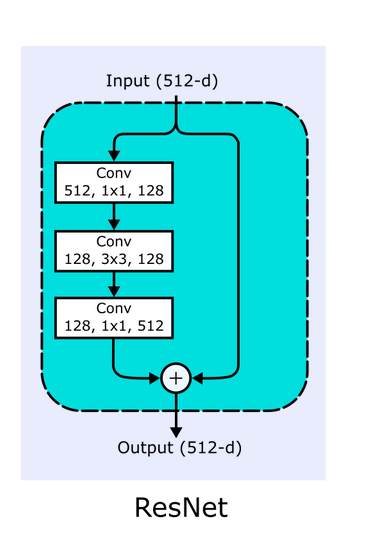
VGG-16 composes of 13 convolutional and 3 fully connected layers with ReLU activation. VGG-16 provides more layers compared to AlexNet and uses smaller filters of 2 × 2 and 3 × 3. It includes 138 million parameters. A deeper version of VGG called VGG-19 is available. VGG-16 is one of the most used architectures in object detection and achieved interesting performances; it’s used for instance in algorithms like Fast R-CNN, Faster R-CNN, SSD

**GoogLeNet**

GoogLeNet is a small network developed in 2014. Their method is different from that of VGGNet and AlexNet. They came up with a new notion known as blocks of inception, where it embeds multi-scale convolutional transformations. The inception block includes filters of varying sizes 1 × 1, 3 × 3 and 5 × 5. It employs a 1 × 1 convolution in the middle of the network to reduce dimensionality and they opted to use global average pooling instead of fully connected layers. The network is made of 22 layers with 5 million parameters. GoogLeNet mainly is used in YOLO object detection model.

### **ResNets**

ResNets are mainly consisting of convolutional and identity blocks. There are many variants of ResNets, for instance, ResNet-34, ResNet-50 which is composed of 26 million parameters, ResNet-101 with 44 million parameters and ResNet-152 which is deeper with 152 layers. ResNet-50 and ResNet-101 are used widely in object detection models. ResNet-101 is used in Faster R-CNN, R-FCN.



### **DarkNet-19**

It is composed of 19 convolutional and 5 max-pooling layers. It uses only 3 × 3 convolutional kernels and several 1 × 1 convolutional kernel to reduce the number of parameters. DarkNet-19 is used in YOLOv2 .

**Swin Transformer** is a type ofVision Transformer. It builds hierarchical feature maps by merging image patches (shown in gray) in deeper layers and has linear computation complexity to input image size due to computation of self-attention only within each local window (shown in red). It can thus serve as a general-purpose backbone for both image classification and dense recognition tasks. In contrast, previous vision Transformers produce feature maps of a single low resolution and have quadratic computation complexity to input image size due to computation of self-attention globally. It is used in one of SOTA object detection networks UniverseNet

**DeciNets** include various models for image classification, object detection, and semantic segmentation and deliver better accuracy-latency tradeoff than any known open-source neural net on the market, including EfficientNets and MobileNets, YoloV5 ( i do not know this network, but this info is shown in a technical comparison)

1. Questions in Software engineering/C++ Programming:

4. Describe a few techniques to do concurrent programming in both C++ and Python. Write some threading implementation samples in C++.

5. Describe the similarities and differences between C++ and Python. How to run C++ scripts in Python and vice versa?

* C++ and Python are programming languages. Both are used for building image processing application
* But C++ is designed to be a compiled language, meaning that it is generally translated into machine language that can be understood directly by the system, making the generated program highly efficient. Whereas, Python is an interpreted language, which means the source code of a Python program is converted into bytecode that is then executed by the Python virtual machine.
* How to run C++ or other application in Python:
  + Use subprocess lib

Example

proc = subprocess.Popen(["./myprog"])

* There are other libs for such implementation

6. Discuss DeepStream and TensorRT. How can we utilize them in Computer Vision products? Can we use more than 2 AI tasks on a Deepstream app?

* DeepStream is a SDK of NVIDIA delivering a complete streaming analytics toolkit for AI-based multi-sensor processing, video, audio and image understanding. It is for vision AI developers, software partners, startups and OEMs building a wide variety of object detection, image classification and instance segmentation application based AI models. Using Deepstream is able to achieve higher accuracy and real-time performance.
* TensorRT: is a machine learning framework that is published by Nvidia to run inference that is machine learning inference on their hardware. TensorRT is highly optimized to run on NVIDIA GPUs. It's likely the fastest way to run a model at the moment.
* Deepstream-app does not support multiple primary gies → can not use more than 2 AI tasks

7. What is the Boost library? When should it be used?

* Boost is a set of libraries for the C++ programming language that provides support for tasks and structures such as linear algebra, pseudorandom number generation, multithreading, image processing, regular expressions, and unit testing
* In our cases, it can be used in multithreading processing because one of the things the standard library doesn't have that boost has is the `shared\_memory` stuff. You can communicate across processes.

8. What is HTTP & Websocket? When should it be used?

* Hypertext Transfer Protocol (HTTP) is an application-layer protocol for transmitting hypermedia documents, such as HTML. It was mainly designed for communication between web browsers and web servers.
* WebSocket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection.
* WebSocket can be used if we want any real-time updated or continuous streams of data that are being transmitted over the network. If we want to fetch old data, or want to get the data only once to process it with an application we should go with HTTP protocol.

9. What are Smart Pointers? List some common types and their applications.

* In modern C++ programming, the Standard Library includes *smart pointers*, which are used to help ensure that programs are free of memory and resource leaks and are exception-safe.
* Common types:
* *unique\_ptr* stores one pointer only. We can assign a different object by removing the current object from the pointer
* By using *shared\_ptr* more than one pointer can point to this one object at a time and it’ll maintain a Reference Counter using *use\_count()* method

10. How to decode a file encrypted with openssl enc -aes-256-cbc -pbkdf2 -iter 121092 -k abc121092 in code ? (Use C++ and openssl library to create a function which has encrypted file path as input, and decrypted file path as output)

Function (){

unsigned char key[EVP\_MAX\_KEY\_LENGTH];

unsigned char iv[EVP\_MAX\_IV\_LENGTH];

unsigned char salt[8]; // openssl tool uses 8 bytes for salt

unsigned char decodeddata[256];

unsigned char ciphertext[256];

unsigned char plaintext[256];

const char \*pass = argv[1]; // use first argument as password (PoC only)

unsigned char \*encodeddata = (unsigned char \*)argv[2]; // use second argument

int decodeddata\_len, ciphertext\_len, plaintext\_len, len;

// Decode base64 string provided as second option

EVP\_ENCODE\_CTX \*ctx;

if(!(ctx = EVP\_ENCODE\_CTX\_new()))

bail();

EVP\_DecodeInit(ctx);

EVP\_DecodeUpdate(ctx, decodeddata, &len, encodeddata, strlen((const char\*)encodeddata));

decodeddata\_len = len;

if(!EVP\_DecodeFinal(ctx, decodeddata, &len))

bail();

EVP\_ENCODE\_CTX\_free(ctx);

// openssl tool format seems to be 'Salted\_\_' + salt + encrypted data

// take it apart

memcpy(salt, decodeddata + 8, 8); // 8 bytes starting at 8th byte

memcpy(ciphertext, decodeddata + 16, decodeddata\_len - 16); // all but the 16 first bytes

ciphertext\_len = decodeddata\_len - 16;

// Get some needed information

const EVP\_CIPHER \*cipher = EVP\_aes\_256\_cbc();

int iklen = EVP\_CIPHER\_key\_length(cipher);

int ivlen = EVP\_CIPHER\_iv\_length(cipher);

int iter = 10000; // default in openssl 1.1.1

unsigned char keyivpair[iklen + ivlen];

// Generate the actual key IV pair

if(!PKCS5\_PBKDF2\_HMAC(pass, -1, salt, 8, iter, EVP\_sha512(), iklen + ivlen, keyivpair))

bail();

memcpy(key, keyivpair, iklen);

memcpy(iv, keyivpair + iklen, ivlen);

// Decrypt data

EVP\_CIPHER\_CTX \*cipherctx;

if(!(cipherctx = EVP\_CIPHER\_CTX\_new()))

bail();

if(!EVP\_DecryptInit\_ex(cipherctx, cipher, NULL, key, iv))

bail();

if(!EVP\_DecryptUpdate(cipherctx, plaintext, &len, ciphertext, ciphertext\_len))

bail();

plaintext\_len = len;

if(!EVP\_DecryptFinal\_ex(cipherctx, plaintext + len, &len))

bail();

plaintext\_len += len;

EVP\_CIPHER\_CTX\_free(cipherctx);

plaintext[plaintext\_len] = '\0'; // add null termination

printf("%s", plaintext);

exit(EXIT\_SUCCESS);

}

}

11. List some commonly used C++ frameworks and libraries.

* OpenCV
* Boost

1. Product Project:

Optional features:

1. One of our promising clients wants to run 4K input using our products. Please show us how this could/should be done?

* I do image scaling before feeding them to the inference phase so 4k input video can be input to our application. However, the accuracy of object detection can be slightly worse because the pedestrian size in the image is too small to detect.

2. Our pipeline could heavily depend on the CPU when processing. How to improve this pipeline? Please show us whether we could use only GPU for processing the video used for the demo.

We have multiple tasks including:

* Image downsizing + pedestrian detection
* Tracking based on pedestrian detection results
* Sending video result to the server

Which can be implemented concurrently by using multithreading to improve this pipeline. But using only GPU make no improvement in this pipeline since GPU only brings advantages upon time-consuming processing for array- based computing. And tracking and sending video to server is not the array-based computing task, except for matrix manipulation.

3. Coding review: Siam Trackers As one of our first Senior Software Engineers, we would like you to have the ability to examine, question and raise any issues in the code when our product comes to the client. We will send you a Github repository of one of our products to assess this ability. This repository will include the C++ code version of a well-known project Pysot (https://github.com/STVIR/pysot). We would like you to give some reviews and assessments of this project. It could be the coding style, documentation, issue tracking, software architecture, building pipeline, and more.

* Comments on coding style:
  + Should input each function the reference of parameters (not parameter value). As code-reader, i can not differ which parameters are writable and which ones are readable
  + Main function in main.cpp is too long, should be refactored.
  + Write comments for each code line. It is difficult for me to understand for what each code-function is implementing.
* Comments on software architecture:
* Create the own data folder for containing the media files, images ….