

[Open in app ↗](#)

Medium



Search



Be part of a better internet. [Get 20% off membership for a limited time](#)

4 Machine Learning Object Detection Solutions



Max Kleiner

6 min read · 1 day ago

[Listen](#)[Share](#)[More](#)

Object Detection APIs and Libraries provides a fast and accurate image object recognition using advanced neural networks developed by machine learning experts and models. It also supports object detection, video detection and object tracking using RetinaNet, YOLOv3 and TinyYOLOv3 pre-trained on datasets.

Solution Script:

https://github.com/maxkleiner/HttpComponent/blob/main/1316_detector25_integrate4solutions.txt

We deliver 4 showcases with the same image to compare and evaluate:

- 1. Integrate Python ImageAI /PyTorch

- 2. THHttpRequestC RestClient / <https://api-ninjas.com/api/objectdetection>
- 3. THHttpConnectionWinInet WinInet API /
https://api.apilayer.com/image_to_text/
- 4. Integrate Python4Delphi / Apilayer

As we can see the algorithm, data and the result is distributable and scalable:

```
Algorithm (model) Data (image) Result (Json) Tech
1. local - local - local python core
2. cloud - local - local/cloud post API
3. cloud - cloud - local/cloud get API
4. cloud - cloud - local rest API
```

The first solution starts with **tiny-yolov3.pt** model from **imagai**:

```
'#using the pre-trained TinyYOL0v3 model,
detector.setModelTypeAsTinyYOL0v3()
detector.setModelPath(model_path)
#loads model path specified above using setModelPath() class method.
detector.loadModel()
custom=detector.CustomObjects(person=True,laptop=True,car=False,train=True, clc
```



S.A.C.M. Elsässische Maschinenbauanstalt Graffenstaden C-Kuppler (2x) — Baujahr 1900



The reference image for the solutions



Result: Start with maXbox5 ImageAI Detector ->
train : 80.25 %
integrate image detector compute ends...

elapsedSeconds:= 4.879268800000 no console attached..

mX5 🚗 executed: 29/07/2024 09:53:49 Runtime: 0:0:8.143 Memload: 75% use

The we asked why the model can't see the persons? It depends on the frame, so by cutting the image (crop) it sees persons but no train anymore!

```
input_path=r"C:\maxbox\maxbox51\examples\1316_elsass_20240728_161420crop.jpg"
```

Result: Start with maXbox5 ImageAI Detector ->

this first line fine

person : 99.29 %

person : 99.58 %

person : 98.74 %

integrate image detector compute ends...

elapsedSeconds:= 4.686975000000 — no console attached..

mX5 🚗 executed: 29/07/2024 10:09:30 Runtime: 0:0:7.948 Memload: 77% use

You can see one false positive in the green bounding box above!

The **Second Solution** is an API from URL_APILAY_DETECT = '<https://api.api-ninjas.com/v1/objectdetection/>';

The Object Detection API provides fast and accurate image object recognition using advanced neural networks developed by machine learning models.

<https://api-ninjas.com/api/objectdetection>

```
const URL_APININ_DETECT= 'https://api.api-ninjas.com/v1/objectdetection/';  
function TestHTTPClassComponentAPIDetection2(AURL, askstream, aApiKey: string)
```

```
function TestHTTPClassComponentAPIDetection2(AURL, askstream, aApiKey: string):  
var HttpReq1: THttpRequestC;  
Body: TMultipartFormBody;  
Body2: TUrlEncodedFormBody; //ct: TCountryCode;
```

```
begin
  Body:= TMultipartFormBody.Create;
  Body.ReleaseAfterSend:= True;
  //Body.Add('code','2','application/octet-stream');
  //Body.AddFromFile('image', exepath+'randimage01.jpg');
  Body.AddFromFile('image',
    'C:\maxbox\maxbox51\examples\1316_elsass_20240728_re

  HttpReq1:= THttpRequestC.create(self);
  httpreq1.userAgent:= USERAGENT3;
  httpReq1.headers.add('X-Api-Key:' +AAPIKEY);
  httpReq1.headers.add('Accept:application/json');
  hhttpReq1.SecurityOptions:= [soSsl3, soPct, soIgnoreCertCNInvalid];
  try
    if HttpReq1.Post1Multipart(AURL, body) then
      result:=HttpReq1.Response.ContentAsString
    else Writeln('APIError '+inttostr(HttpReq1.Response.StatusCode2));
  finally
    writeln('Status3: '+gethtpcod(HttpReq1.Response.statuscode2))
    HttpReq1.Free;
    sleep(200)
    // if assigned(body) then body.free;
  end;
end;
```

This result is a post from a multipartform body stream and you need an API key, then the result is a JSON back, as you can see, we need a call to

HttpReq1.Post1Multipart for uploading files.:

POST data using the Content-Type multipart/form-data

JSON

```
1  [
2    {
3      "label": "train",
4      "confidence": "0.76",
5      "bounding_box": {
6        "x1": "-6",
7        "y1": "291",
8        "x2": "1173",
9        "y2": "1347"
10      }
11    },
12    {
13      "label": "person",
14      "confidence": "0.72",
15      "bounding_box": {
16        "x1": "535",
17        "y1": "854",
18        "x2": "815",
19        "y2": "1519"
20      }
21    },
22    {
```

The **third** solution wants to get the text back from the image. The Image to Text API detects and extracts text from images using state-of-the-art optical character recognition (OCR) algorithms. It can detect texts of different sizes, fonts, and even handwriting on pictures or draws.

```
URL_APILAY_IMG2TEXT = 'https://api.apilayer.com/image_to_text/url?url=%s'; func
```

```

URL_APILAY_IMG2TEXT = 'https://api.apilayer.com/image_to_text/url?url=%s';

function Image_to_text_API2(AURL, url_imgpath, aApiKey: string): string;
var httpq: THttpConnectionWinInet;
    rets: TStringStream;
    heads: TStrings; iht: IHURLConnection; //localhost:HTTPConnectionLostEvent;
begin
    httpq:= THttpConnectionWinInet.Create(true);
    rets:= TStringStream.create('');
    heads:= TStringlist.create;
    try
        heads.add('apikey=' + aAPIkey);
        iht:= httpq.setHeaders(heads);
        httpq.Get(Format(AURL,[url_imgpath]), rets);
        if httpq.getresponsecode=200 Then result:= rets.datastring
        else result:='Failed:' +
            itoa(Httpq.getresponsecode)+Httpq.GetResponseHeader('message');
    except
        writeln('EWI_HTTP: '+ExceptionToString(exceptiontype,exceptionparam));
    finally
        httpq:= Nil;
        heads.Free;
        rets.Free;
    end;
end;           //}

```



And the model is able to read the **name** of the Locomotive!:

Result_: {"lang":"en","all_text":"18130\n BERTHOLD","annotations":["18130","BERTHOLD"]}

mX5 🚂 executed: 29/07/2024 11:04:12 Runtime: 0:0:3.527 Memload: 81% use

The **forth** and last solution in this machine learning package is a Python one as in Python for maXbox or Python4Delphi available:

```

procedure PyCode(imgpath, apikey: string); begin with TPythonEngine.Create(Nil)

```



```

procedure PyCode(imgpath, apikey: string);
begin

```

```
with TPythonEngine.Create(Nil) do begin
//pythonhome:= 'C:\Users\User\AppData\Local\Programs\Python\Python312\' ;
try
  loadDLL;
  autofinalize:= false;
  ExecString('import requests, sys');
  ExecStr('url= "https://api.apilayer.com/image_to_text/url?url=' + imgpath + "'");
  ExecStr('payload = {}');
  ExecStr('headers= {"apikey": "' + apikey + '"}');
  Println(EvalStr('requests.request("GET",url,headers=headers, data=payload)');
  Println('Version: ' + EvalStr('sys.version')));
except
  raiseError();
finally
  free;
end;
end;
end;
```

{"lang": "en", "all_text": "18130\\nBERTHOLD", "annotations": ["18130", "BERTHOLD"]}

Version: 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]

mX5  executed: 29/07/2024 11:18:13 Runtime: 0:0:4.60 Memload: 79% use



S.A.C.M. Elsässische Maschinenbauanstalt Graffenstaden C-Kuppler (2x) – Baujahr 1900

Conclusion and Summary

1. Built with simplicity in mind, **ImageAI** supports a list of state-of-the-art Machine Learning algorithms for image prediction, custom image prediction, object detection, video detection, video object tracking and image predictions trainings. ImageAI currently supports image prediction and training using 4 different Machine Learning algorithms trained on the ImageNet-1000 dataset. ImageAI also supports object detection, video detection and object tracking using RetinaNet, YOLOv3 and TinyYOLOv3 trained on COCO dataset. Finally, ImageAI allows you to train custom models for performing detection and recognition of new objects.

<https://github.com/OlafenwaMoses/ImageAI>

2. Object Detection API — The **Object Detection API** provides fast and accurate image object recognition using advanced neural networks developed by machine learning experts. It also has a Live Demo or rules with Mime Post Multipart FormData_:

<https://api-ninjas.com/api/objectdetection>

<https://github.com/maxkleiner/HttpComponent>

3. Recognizes and reads the text embedded in images very accurate and usable. **Image to Text API** uses a neural net (LSTM) based OCR engine which is focused on line recognition, but also supports recognizing the character patterns. It supports both handwriting and printed materials.

It will extract the text information easily, even though the text or number is positioned with angle, like Berthold.

https://apilayer.com/marketplace/image_to_text-api

4. The **Requests library** in Python is one of the integral parts of Python for making HTTP requests to a specified URL as post or get. Whether it be REST APIs or Web Scraping, requests are a must to be learned for proceeding further with these technologies.

5. Out of the examples above but also mention it: The **Face Detect API** uses state of the art computer vision algorithms to accurately and efficiently detect faces in images.

<https://api-ninjas.com/api/facedetect>

The screenshot shows the API Ninjas website with the URL <https://api-ninjas.com/api/facedetect>. The main content is titled "Face Detect API" and describes how it uses state-of-the-art computer vision algorithms to detect faces in images. It includes a "Sample Request" section with a "Live Demo" button, an "Upload Image" input field, and a "Browse" button. A sample image of two people standing next to a steam locomotive is displayed. On the left sidebar, there's a "API Directory" section with various categories like Finance, Internet/Technology, and more.

Interface for The Face Detect API

Originally published at <http://softwareschule.code.blog> on July 29, 2024.

Machine Learning

Object Detection

API

Api Development



Edit profile

Written by Max Kleiner

27 Followers

Max Kleiner's professional environment is in the areas of OOP, UML and coding - among other things as a trainer, developer and consultant.

More from Max Kleiner

The screenshot shows a web browser displaying the API-ninjas.com object detection API documentation. The URL in the address bar is <https://api-ninjas.com/api/objectdetection>. The page lists various endpoints and their descriptions. On the left, there's a sidebar with links to other APIs like Aircraft, Airlines, Airports, etc. The main content area shows the '/v1/objectdetection' endpoint, which is an HTTP POST method. It describes the API as returning a list of detected objects, confidence percentages, and bounding boxes, with a threshold of 0.3. It requires an 'image' file (JPEG or PNG) and an 'X-Api-Key'. A sample request URL is provided: <https://api.api-ninjas.com/v1/objectdetection>. Below the URL are two buttons: 'Upload Image' and 'Browse'. An image of a person walking away from a large tree in a field is shown as a sample input.

 Max Kleiner in Nerd For Tech

Object Detection API

We call it AIM and this stands for Artificial Intelligence Machine.

Apr 27  4



 Ronak Patel in Nerd For Tech

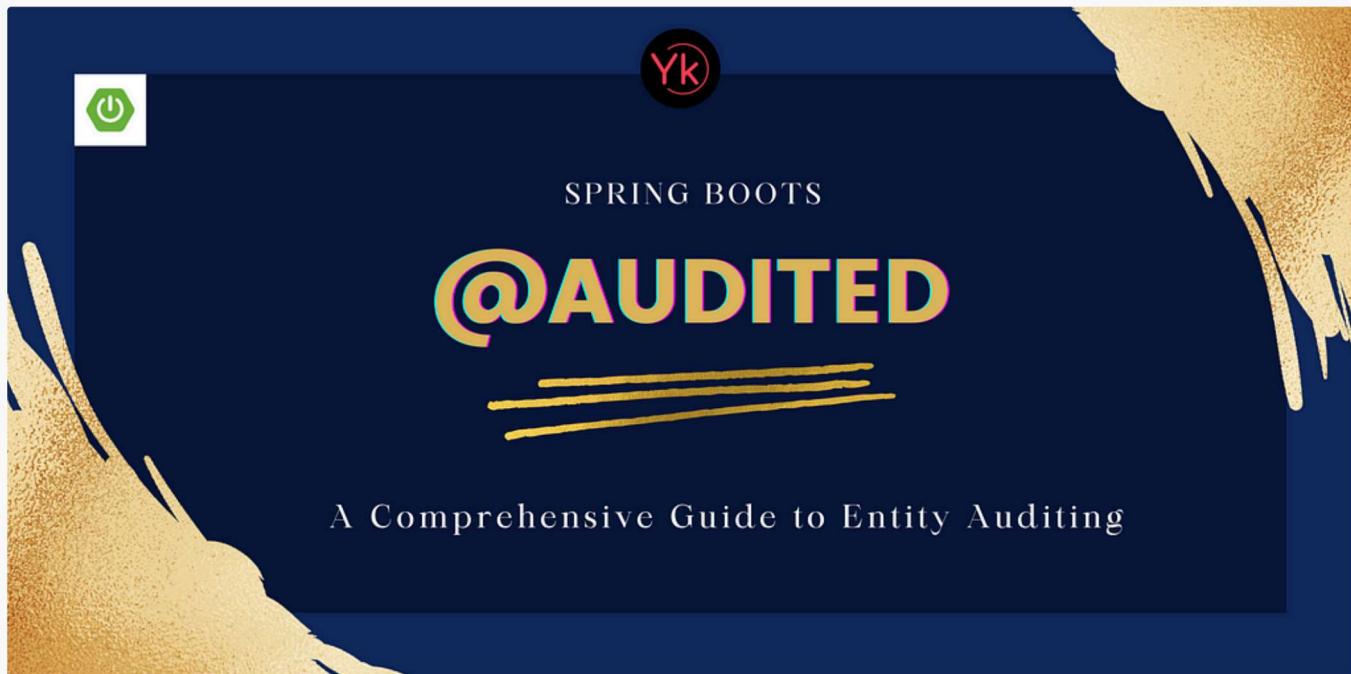
Best 5 Micro Front-end Frameworks Every Developer Should Know

In today's dynamic landscape of web development, the demand for scalable, agile and maintainable solutions has never been more pressing. As...

Apr 18 226 3



...



 Yeran Kods in Nerd For Tech

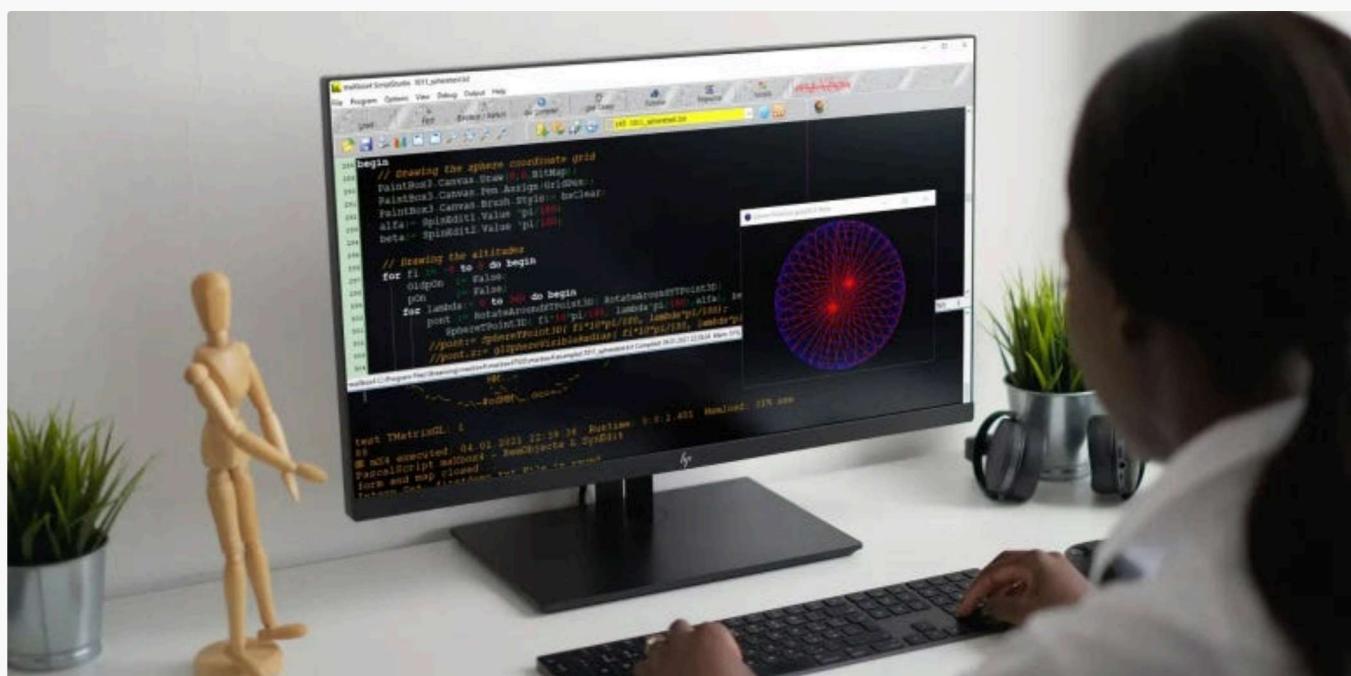
@Audited Annotation in Spring Boot

Enhance Your Spring Boot Applications with @Audited for Robust Data Auditing

Feb 9 83



...





Max Kleiner

Train a Logic Classifier

The model below this article depicts the architecture for a multilayer perceptron network designed specifically to solve the XOR problem...

Feb 16, 2021 12

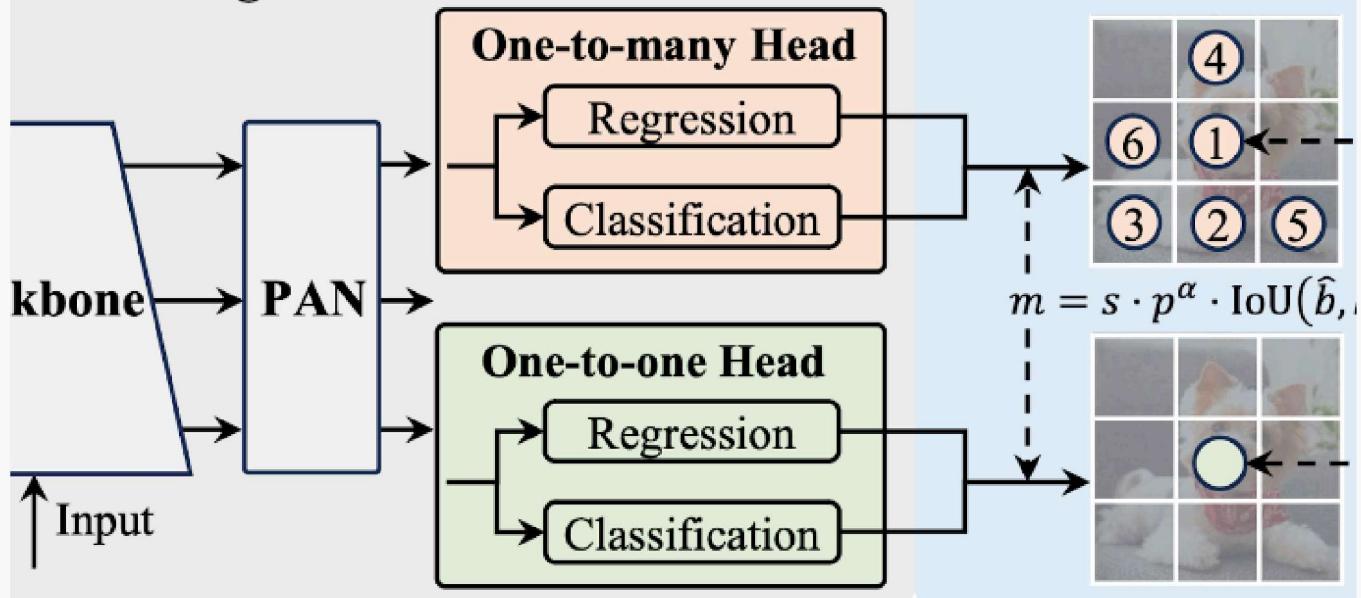


...

See all from Max Kleiner

Recommended from Medium

Label Assignments



TUC The useless channel

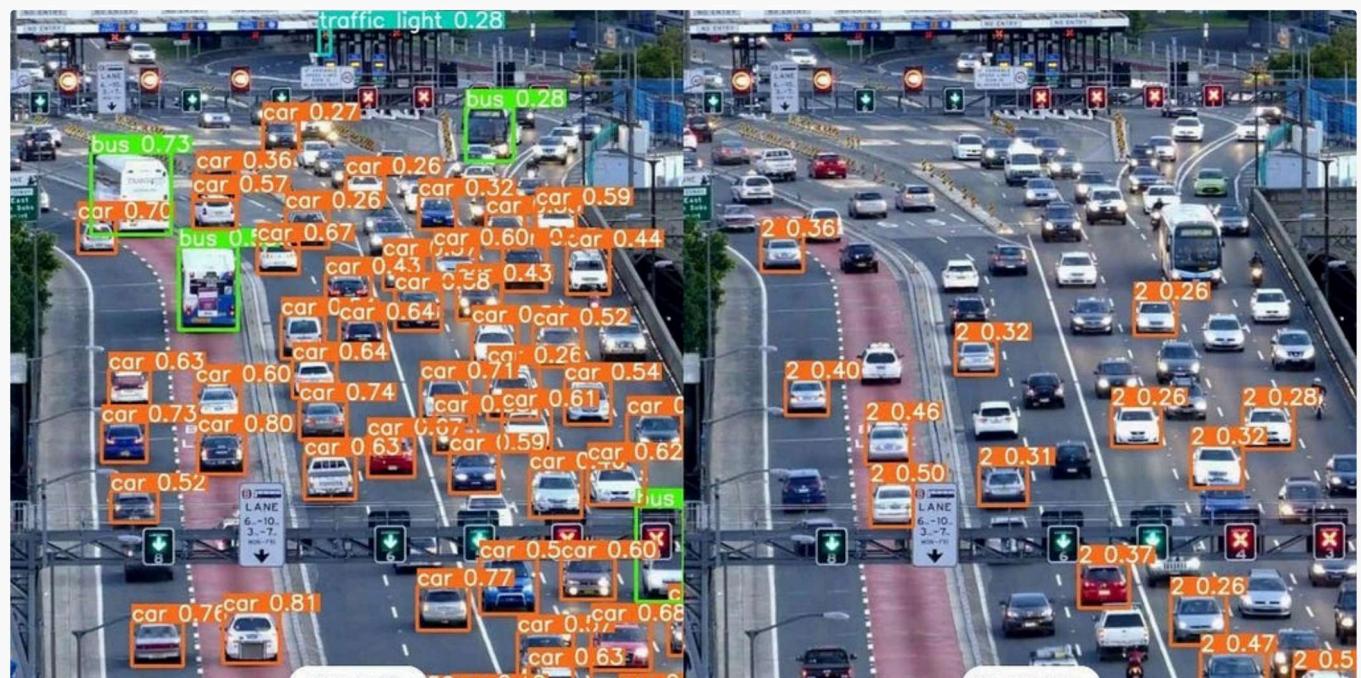
YOLO V10

YOLOv10, the latest version of the YOLO series, introduces several key innovations and improvements over its predecessors. Here are the...

Jun 20 22



...



Daniel García

The Next Generation of Object Detection is Here: YOLOv10

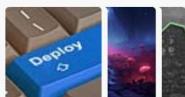
Discover How YOLOv10 Redefines Real-Time Object Detection with Unmatched Speed and Precision

May 27 53 1



...

Lists



Predictive Modeling w/ Python

20 stories · 1413 saves



Practical Guides to Machine Learning

10 stories · 1713 saves



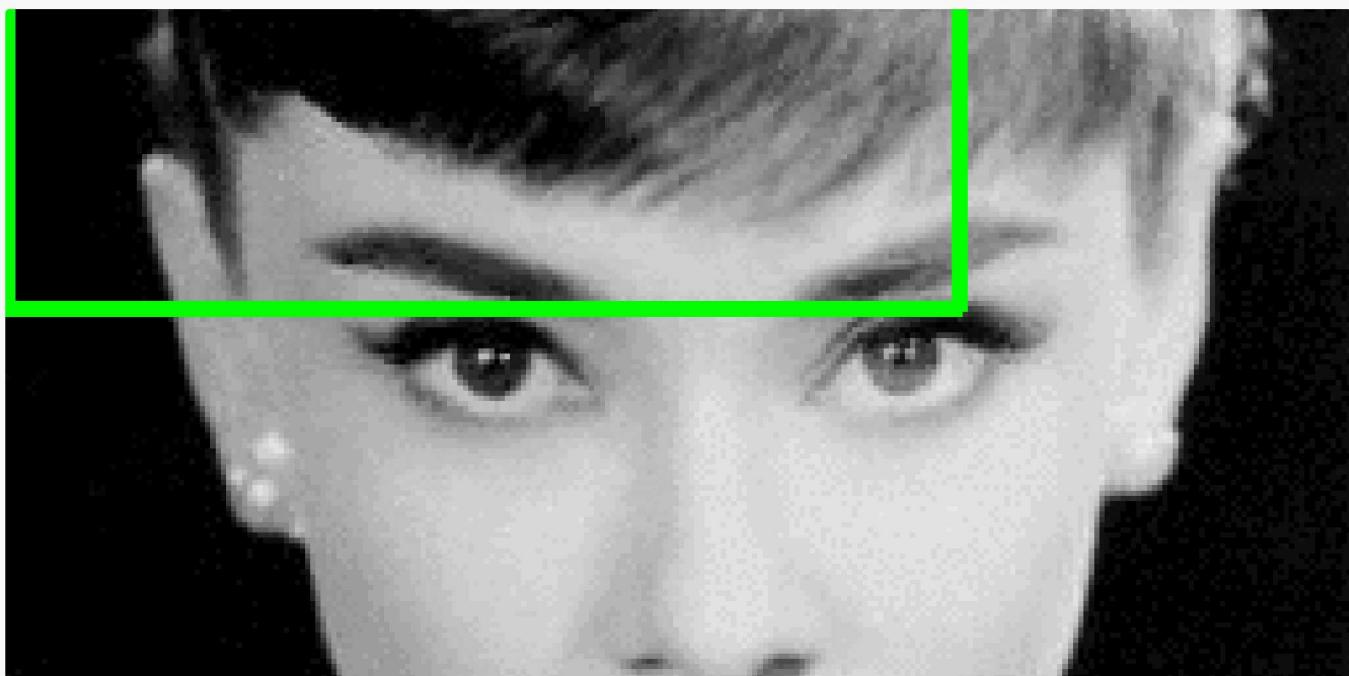
Natural Language Processing

1607 stories · 1169 saves



The New Chatbots: ChatGPT, Bard, and Beyond

12 stories · 431 saves



pju

Object detection: Convolutional Implementation of Sliding Windows

In the previous chapter, we've studied about Sliding Windows and their concept. Their most disadvantages are Computational Cost, now we...

5d ago 1



...

```
*__, a, b, *__ = [1, 2, 3, 4, 5, 6]
print(__, __)
```

What does this print?

- A) Syntax error
- B) [1] [4, 5, 6]
- C) [1, 2] [5, 6]
- D) [1, 2, 3] [6]
- E) <generator object <genexpr> at 0x1003847c0>

 Liu Zuo Lin

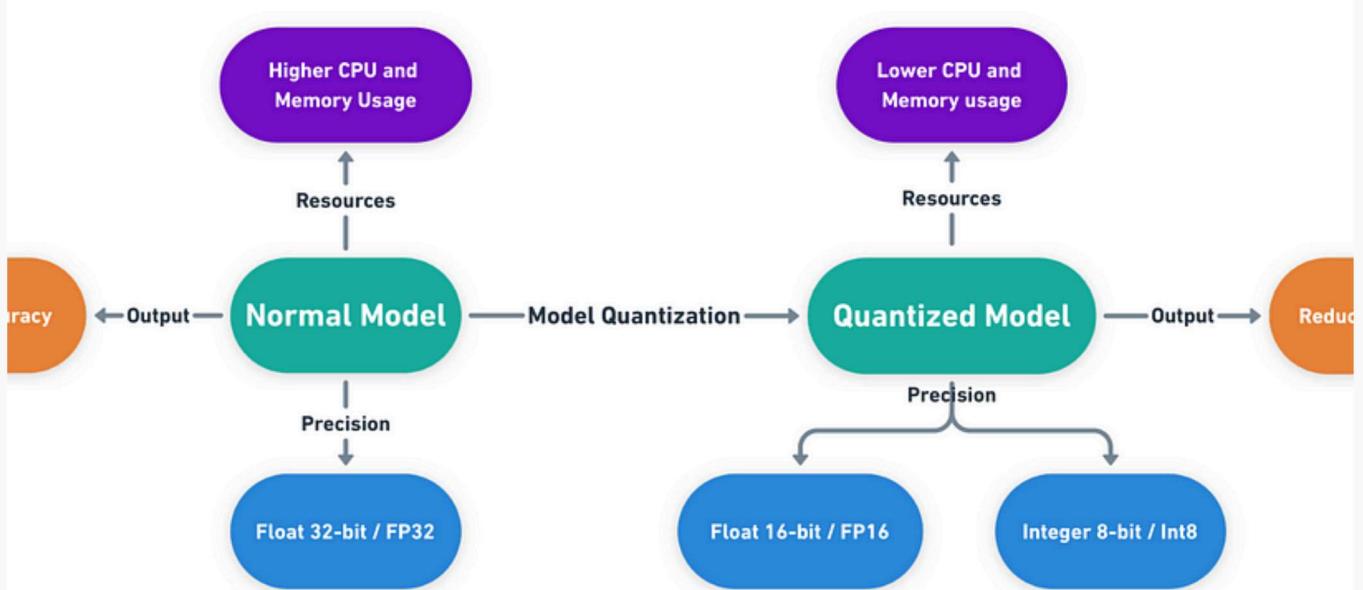
You're Decent At Python If You Can Answer These 7 Questions Correctly

No cheating pls!!

★ Mar 6 ⌘ 6.3K 🗣 30



...



 sulav shrestha

Quantizing YOLO v8 models

In today's world, artificial intelligence is everywhere—from our smartphones to smart homes. But how do these powerful AI models fit...

Jul 2 18



...

AMAZON.COM**Software Development Engineer**

Seattle, WA

Mar. 2020 – May 2021

- Developed Amazon checkout and payment services to handle traffic of 10 Million daily global transactions
- Integrated Iframes for credit cards and bank accounts to secure 80% of all consumer traffic and prevent CSRF, cross-site scripting, and cookie-jacking
- Led Your Transactions implementation for JavaScript front-end framework to showcase consumer transactions and reduce call center costs by \$25 Million
- Recovered Saudi Arabia checkout failure impacting 4000+ customers due to incorrect GET form redirection

Projects**NinjaPrep.io (React)**

- Platform to offer coding problem practice with built in code editor and written + video solutions in React
- Utilized Nginx to reverse proxy IP address on Digital Ocean hosts
- Developed using Styled-Components for 95% CSS styling to ensure proper CSS scoping
- Implemented Docker with Seccomp to safely run user submitted code with < 2.2s runtime

HeatMap (JavaScript)

- Visualized Google Takeout location data of location history using Google Maps API and Google Maps heatmap code with React
- Included local file system storage to reliably handle 5mb of location history data
- Implemented Express to include routing between pages and jQuery to parse Google Map and implement heatmap overlay



Alexander Nguyen in Level Up Coding

The resume that got a software engineer a \$300,000 job at Google.

1-page. Well-formatted.

Jun 1

15.2K

235



...

[See more recommendations](#)