

پروژه 26 میلیون دلاری

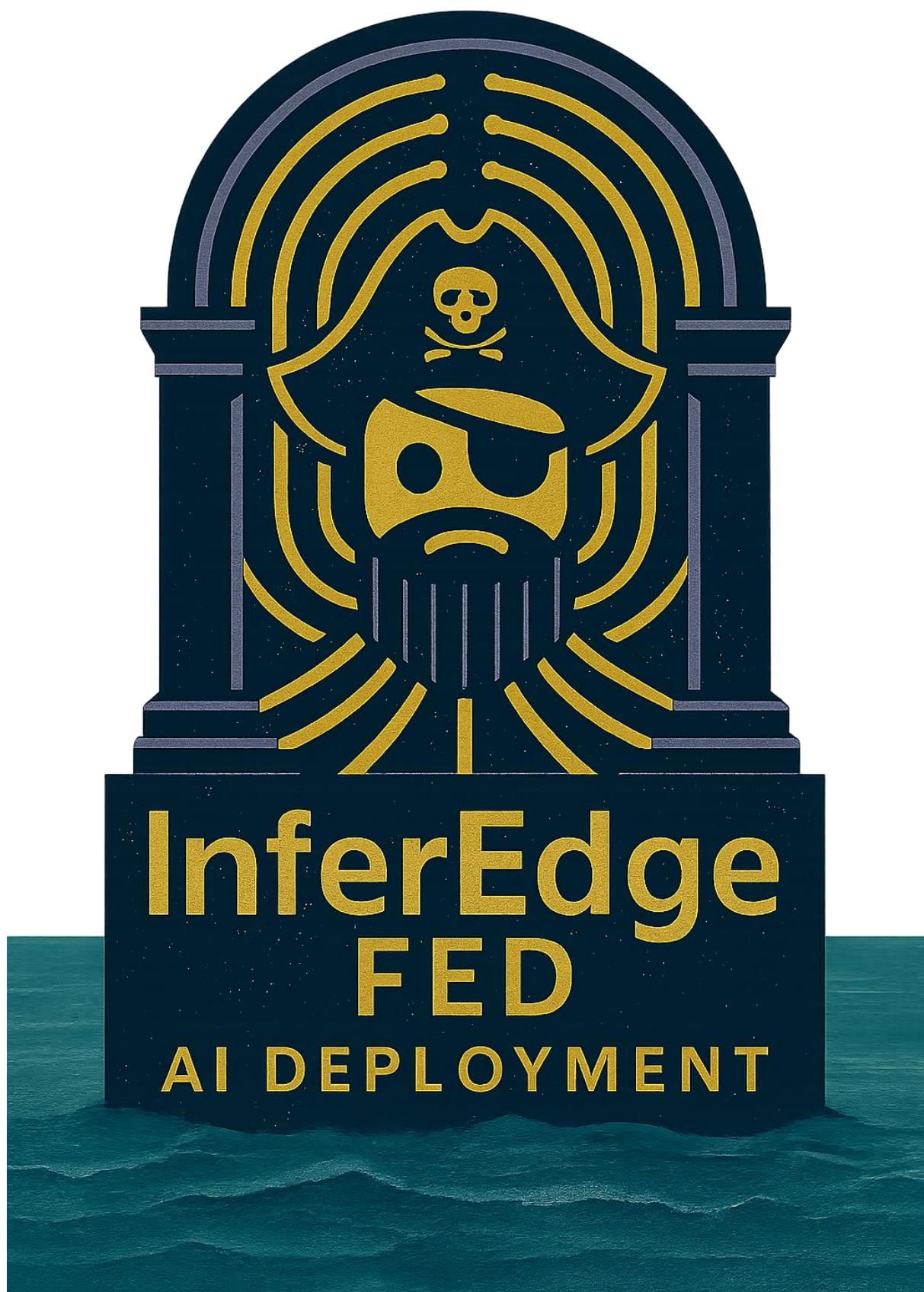


**InferEdge
FED**

توسعه هوش مصنوعی غیرمتırکز

خصوصی

『YouTube』(<https://youtu.be/v1eVU-esCjQ>)



ترجمه صوتی مادری با نروشن شخصی یا سفارشی ماموریت ۱

 کتابخانه‌های مورد نیاز برای اجرای کد در **ویندوز**

Vosk .1 موتور تشخیص گفتار آفلاین

```
py -m pip install vosk
```

sounddevice .2 دریافت صوت از میکروفون

```
py -m pip install sounddevice
```

 **Vosk** مدل فارسی

دانلود مدل فارسی از این لینک:

<https://alphacepheli.com/vosk/models/vosk-model-small-fa-0.4.zip>

فایل را **unzip** کن و مسیرش را در کد مشخص کن:

```
model_path = r"C:\مسیر\vosk-model-small-fa-0.4"
```

```

❸ speech_text.py > ...
1  import sounddevice as sd
2  import queue
3  import vosk
4  import json
5
6  model_path = r"C:\2025\meet-class\vosk-model-small-fa-0.4\vosk-model-small-fa-0.4"
7  model = vosk.Model(model_path)
8
9  output_file = "transcript.txt"
10 q = queue.Queue()
11
12 def callback(indata, frames, time, status):
13     q.put(bytes(indata))
14
15 with open(output_file, "a", encoding="utf-8") as f:
16     with sd.RawInputStream(samplerate=16000, blocksize=8000, dtype='int16',
17                            channels=1, callback=callback):
18         rec = vosk.KaldiRecognizer(model, 16000)
19         print("🎧 Listening...")
20
21     while True:
22         data = q.get()
23         if rec.AcceptWaveform(data):
24             result = json.loads(rec.Result())
25             text = result.get("text", "").strip()
26             if text:
27                 print("📄", text)
28                 f.write(text + "\n")
29                 f.flush()
30             else:
31                 partial = json.loads(rec.PartialResult())
32                 partial_text = partial.get("partial", "").strip()
33                 if partial_text:
34                     print("🤖", partial_text)
35

```

[Github: Fedia/speech_text.py at main · dewebedes/Fedia](https://github.com/Fedia/speech_text.py)

Ollama + DeepSeek-R1 نصب و اجرای

نصب Ollama روی ویندوز

- دانلود از : <https://ollama.com/download>

- نصب و اجرا :

```
ollama serve
```

DeepSeek-R1 مدل دریافت 2.

```
ollama pull deepseek-r1:7b
```

این مدل مددود 13.2 گیگابایت حجم دارد و برای پاسخگویی، ترجمه، و reasoning مناسب است

3. اجرای مدل در شل

```
ollama run deepseek-r1:7b
```

سپس می‌توانی مستقیماً با مدل چت کنی، مثلاً:

سلام، حال شما چطور است؟

اجرای معادل در پایتون با subprocess

```
import subprocess
```

```
def run_ollama(prompt):  
    command = [  
        "ollama", "run", "deepseek-r1:7b"  
    ]  
  
    process = subprocess.Popen(command, stdin=subprocess.PIPE,  
                             stdout=subprocess.PIPE, stderr=subprocess.PIPE, text=True)
```

```
output, error = process.communicate(prompt)  
return output.strip()
```

#نمونه استفاده

```
response = run_ollama("Translate this Persian sentence to English:  
سلام، حال شما چطور است؟")
```

```
print("🧠 پاسخ مدل:", response)
```

نکات مهم

- قبل از اجرای پایتون، مطمئن شو که **ollama serve** در پس زمینه فعال باشد
- می توانی از این تابع برای ترجمه، پاسخگویی، یا هرگز تزریق پرامپت های نمادین استفاده کنی

صدای فارسی—Silero TTS (اچکا) فینگلیش



کتابخانه‌های مورد نیاز



کتابخانه

کاربرد

torch

بارگذاری و اجرای مدل TTS

soundfile

ذخیره‌ی خروجی صوتی به فرمت WAV

numpy

تبدیل و نرمال‌سازی داده‌های صوتی

os

مدیریت فایل‌ها و پوشه‌ها

warnings

مذف هشدارهای غیرضروری در اجرا

نصب سریع:

pip install torch soundfile numpy

عملکرد کد نریشن



۱. بارگذاری مدل Silero TTS با زبان انگلیسی و صدای پیش‌فرض (لیز)

۲. فواندن فطوط متنی از فایل en.txt

۳. تبدیل هر فط به گفتار با استفاده از مدل

۴. نرمال‌سازی و ذخیره‌ی فایل صوتی به صورت ... line_1.wav, line_2.wav, ...

۵. پخش فودکار فایل صوتی پس از تولید

فروجی‌ها در پوشه‌ی narration_output ذخیره می‌شوند

محدودیت مدل Silero برای زبان فارسی

مدل (سمی Silero TTS) در حال حاضر فقط از زبان‌های زیر پشتیبانی می‌کند:

[en', 'ru', 'de', 'es', 'fr', 'ua', 'tt', 'uz', 'ba', 'xal', 'indic', 'cyrillic', 'multi']

زبان فارسی (fa) در لیست پشتیبانی شده نیست، و تلاش برای بازگذاری آن باعث خطای AssertionError می‌شود.

(اهم) خلاصه: با پس محدودیت فارسی با فینگلیش

برای تولید صدای فارسی بدون مدل فارسی، از ووش زیر استفاده می‌کنیم:

1. ترجمه‌ی متن فارسی به فینگلیش

مثال:

→ salam, halet chetore؟؟

3. استفاده از مدل انگلیسی برای نریشن فینگلیش

چون مدل انگلیسی تلفظ فینگلیش را با لهجه‌ی قابل قبول تولید می‌کند، خروجی صوتی شبیه فارسی شنیده می‌شود

4. ترکیب با ترجمه فودکار

می‌توان از DeepSeek برای تبدیل فارسی به فینگلیش استفاده کرد و سپس نریشن گرفت

[GitHub: Fedia/narration.py at main · dewebdes/Fedia](https://github.com/dewebdes/Fedia/blob/main/narration.py)

اعمال فیلتر صوتی با SO-vits-SVC

فیلتر صوتی (Voice Conversion) قادر است صدای تولیدشده را به صدای دلمواه تبدیل کند — متن اگر آن صدا فقط از یک سفرانی یوتیوب استخراج شده باشد.

با استفاده از ابزارهایی مانند:

- <https://github.com/innnky/so-vits-svc.git>
- `ffmpeg`
- `yt-dlp`

می‌توان صدای خام را از ویدیو استخراج کرد، آن را به قطعات ۲۰ ثانیه‌ای تقسیم کرد، و سپس مدل گفتوگویی فیلتر صوتی را آموزش داد.

(اهمیت برای منابع محدود: 

در صورت نداشتن GPU قدرتمند، می‌توان از نوتبوک‌های آماده در Colab یا Kaggle استفاده کرد، مانند:

<https://www.kaggle.com/code/shangguanmingtian/so-vits-svcnotebook1-1>

آموزش مدل

ماموریت: ۲

آموزش مدل زبان از پایه با NanoGPT

ساخت و آموزش یک مدل زبان کوچک (GPT) مانند گسترش به مدل‌های بزرگ‌تر، و استفاده از موازی‌سازی برای بهینه‌سازی منابع.

ابزارهای پایه

ابزار

کاربرد

[NanoGPT](#)

چارچوب سبک برای آموزش GPT از صفر

[PyTorch](#)

کتابخانه اصلی برای مدل‌سازی و آموزش

[DDP](#)

موازی‌سازی داده‌ها (روی چند GPU)

[FSDP](#)

موازی‌سازی کامل پارامترها برای مدل‌های بزرگ

[jkbhagatio/nanoGPT](#)

نسخه توسعه‌یافته NanoGPT با پشتیبانی از FSDP و DDP

نتیجه‌گیری

با NanoGPT و موازی‌سازی DDP/FSDP، می‌توان یک مدل زبان فارسی از پایه آموزش داد — قابل اجرا روی GPU شفمی یا نوت‌بوک‌های کلود، و قابل ترکیب با نریشن و فیلتر صوتی برای تولید صدای نمادین و قابل فهم.