



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

تحقیقات درس برنامه سازی وب

عنوان:

معرفی Spring AI و مدل های تبدیل متن به تصویر

نگارش

امیراردلان دهقان پور ۴۰۱۱۰۵۹۰۱

امیدرضا معصومی ۴۰۱۱۰۶۵۲۲

بهمن ۱۴۰۳

۱ ضرورت استفاده از ابزارهای هوش مصنوعی در توسعه برنامه های وب

در دنیای مدرن امروزی، استفاده از هوش مصنوعی به عنوان ابزاری مؤثر در بهبود فرآیندهای توسعه نرم افزار و تولید محتوا به شدت گسترش یافته است. یکی از چالش های مهم در توسعه نرم افزارهای تحت وب، تولید خودکار محتوای بصری متناسب با ورودی های کاربران است. در بسیاری از کاربردها مانند طراحی گرافیکی، تبلیغات، رسانه های اجتماعی، آموزش و تجارت الکترونیک، کاربران نیاز دارند که تنها با وارد کردن یک توضیح متنی، تصویری متناسب با آن دریافت کنند. این فرایند، اگر به صورت دستی انجام شود، نیاز به زمان، تخصص و منابع مالی زیادی دارد.

از طرف دیگر، پیشرفت های اخیر در مدل های مولد Generative AI امکان تبدیل متن به تصویر text-to-image را فراهم کرده است. مدل هایی مانند Stable Diffusion ، Dall-e و Midjourney قادرند بر اساس توضیحات متنی، تصاویری منحصر به فرد و واقع گرایانه تولید کنند. با این حال، یک چالش اساسی برای توسعه دهندگان، ادغام این مدل های هوش مصنوعی در برنامه های تحت وب است. بسیاری از این مدل ها به منابع پردازشی قوی نیاز دارند و استفاده از آنها مستلزم مدیریت مناسب درخواست ها، بهینه سازی هزینه ها و اطمینان از کیفیت و امنیت خروجی ها است.

راهکار ما استفاده از Spring AI در Spring Boot است. Spring AI بستری را فراهم می کند که بتوان به سادگی مدل های هوش مصنوعی را در سرویس های وب پیاده سازی کرد. این ابزار امکان ادغام آسان API ها با هوش مصنوعی را فراهم کرده و چالش های مربوط به مدیریت درخواست ها و امنیت داده ها را کاهش می دهد. از طریق Spring AI، می توان با فراخوانی یک REST API، ورودی های متنی کاربران را دریافت و آنها را به یک مدل تبدیل متن به تصویر ارسال کرد. سپس خروجی تصویر تولید شده، به کاربر بازگردانده می شود.

هدف اصلی ما در این تحقیق، بررسی نحوه استفاده از Spring AI برای پیاده سازی قابلیت تبدیل متن به تصویر در یک برنامه تحت وب است. ما نشان خواهیم داد که چگونه می توان با حداقل کدنویسی، از قدرت مدل های پیشرفته هوش مصنوعی در Spring Boot بهره برد و آنها را در قالب یک سرویس وب ارائه کرد.

۲ معرفی Spring AI و مدل‌های تبدیل متن به تصویر

Spring AI یک کتابخانه‌ی نسبتاً جدید در Spring Boot است که با هدف ساده‌سازی ادغام هوش مصنوعی در برنامه‌های جاوا طراحی شده است [۱]. این ابزار به توسعه‌دهندگان امکان می‌دهد که بدون نیاز به مدیریت مستقیم مدل‌های پیچیده‌ی یادگیری عمیق، از قابلیت‌های AI بهره ببرند. Spring AI از مدل‌های مختلف ارائه‌دهندگان AI مانند OpenAI API، Hugging Face Inference، AWS Bedrock و پشتیبانی می‌کند [۲].

مزایای Spring AI

- یکپارچگی ساده با Spring Boot: توسعه‌دهندگان می‌توانند به راحتی این کتابخانه را در پروژه‌های خود اضافه کنند و از آن در کنترلرها و سرویس‌های REST استفاده کنند.
- پشتیبانی از مدل‌های مختلف: می‌توان از مدل‌های آماده‌ی پردازش زبان طبیعی (NLP) و همچنین مدل‌های تولید تصویر بهره برد.

در حال حاضر، Spring AI از سرویس‌های مختلفی برای پردازش و تولید تصویر پشتیبانی می‌کند. برخی از مهم‌ترین مدل‌های تبدیل متن به تصویر که در Spring AI قابل استفاده هستند، شامل موارد زیر می‌شوند:

- DALL·E 2: مدل پیشرفته‌ی توسعه‌یافته توسط OpenAI که قادر است تصاویر باکیفیت بالا را از توضیحات متنی تولید کند [۳].
- Stable Diffusion: مدلی که توسط Stability AI ارائه شده و امکان اجرای محلی آن نیز وجود دارد [۴].
- Deep Floyd IF: مدل تولیدی از Deep Mind که توانایی تولید تصاویر با جزئیات بالا را دارد [۵].

نحوه‌ی کارکرد Spring AI در تبدیل متن به تصویر

۱. ارسال درخواست از طریق REST API: کاربر یک متن توضیحی (Prompt) را به سرور ارسال می‌کند.

۲. پردازش درخواست: Spring AI این درخواست را به یکی از مدل‌های Text-to-Image هدایت می‌کند.

۳. دریافت و بازگشت نتیجه: مدل موردنظر تصویر تولید شده را بازمی‌گرداند و سرور آن را برای نمایش به کاربر ارسال می‌کند.

با استفاده از این قابلیت‌ها، توسعه‌دهندگان می‌توانند بدون نیاز به دانش عمیق در زمینه‌ی یادگیری ماشین، از قدرت مدل‌های مولد تصویر در برنامه‌های تحت وب خود استفاده کنند. در ادامه، یک نمونه‌ی پیاده‌سازی در Spring Boot ارائه خواهد شد.

۳ چگونگی کارکرد و مثال آن

در این پیاده‌سازی، ابتدا کاربر یک متن توصیفی را به عنوان ورودی وارد می‌کند. این متن از طریق یک REST API که در Spring Boot پیاده‌سازی شده است، به Stability AI ارسال می‌شود. این سرویس از مدل‌های یادگیری عمیق برای تبدیل متن به تصویر استفاده کرده و نتیجه را در قالب یک تصویر رمزگذاری‌شده بازمی‌گرداند. سپس Spring Boot این تصویر را از حالت Base64 خارج کرده و به عنوان پاسخ به درخواست ارسال می‌کند. این فرایند شامل مراحل زیر است:

۱. دریافت متن توصیفی از کاربر از طریق POST API.
۲. ارسال درخواست به Stability AI همراه با کلید مجوز API.
۳. دریافت پاسخ که شامل تصویر رمزگذاری‌شده است.
۴. تبدیل تصویر از Base64 به فرمت قابل نمایش مانند PNG.
۵. ارسال تصویر به کاربر به عنوان پاسخ HTTP.

۴ پیاده‌سازی مثال

در این مثال، از Stability AI برای تبدیل متن به تصویر استفاده شده است. برای ارسال درخواست، ابتدا یک API Key معتبر دریافت شده و سپس یک کنترلر در Spring Boot ایجاد شده است.

۱-۴ کنترلر Spring Boot

در این بخش درخواست را به کمک پایانه تعریف شده دریافت می کنیم و سپس prompt مرتبط با دستور را از داخل آن استخراج می کنیم و سپس آن را ارسال می کنیم و همچنین کلید خصوصی خود را نیز در داخل header قرار می دهیم.

```
no usages
@PostMapping("/imagegen")
public ResponseEntity<byte[]> generateImage(@RequestBody ImageRequest request) {
    RestTemplate restTemplate = new RestTemplate();

    MultiValueMap<String, Object> body = new LinkedMultiValueMap<>();
    body.add("prompt", request.prompt);

    HttpHeaders headers = new HttpHeaders();
    headers.set("Authorization", "Bearer " + API_KEY);
    headers.setContentType(MediaType.MULTIPART_FORM_DATA);
    headers.setAccept(Collections.singletonList(MediaType.APPLICATION_JSON));

    HttpEntity<MultiValueMap<String, Object>> entity = new HttpEntity<>(body, headers);
```

شکل ۱: نمونه کد کنترلر در Spring Boot

```
HttpEntity<MultiValueMap<String, Object>> entity = new HttpEntity<>(body, headers);

try {
    ResponseEntity<Map> response = restTemplate.exchange(API_URL, HttpMethod.POST, entity, Map.class);
    String base64Image = (String) response.getBody().get("image"); // Extract base64 image

    // Convert base64 to byte array
    byte[] imageBytes = Base64.getDecoder().decode(base64Image);

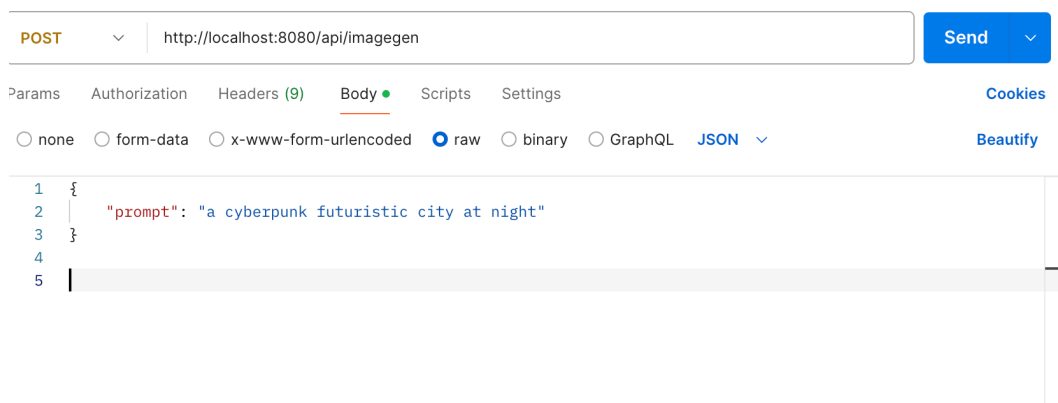
    // Return as an image file
    HttpHeaders imageHeaders = new HttpHeaders();
    imageHeaders.setContentType(MediaType.IMAGE_PNG);

    return new ResponseEntity<>(imageBytes, imageHeaders, HttpStatus.OK);
} catch (Exception e) {
    return ResponseEntity.status(500).body(("Error generating image: " + e.getMessage()).getBytes());
}
```

شکل ۲: ادامه کد کنترلر

۲-۴ ارسال درخواست به API

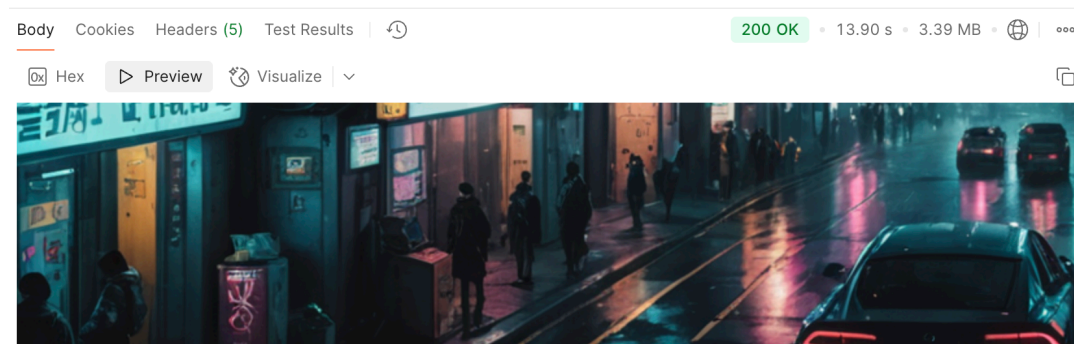
درخواست باید به صورت POST ارسال شود. نمونه‌ای از داده‌های ارسالی در Postman در صفحه بعد آمده است. است:



شکل ۳: نمونه درخواست POST در Postman

۳-۴ مشاهده خروجی

مشاهده می کنیم که پس از ارسال درخواست خروجی آن در بخش پاسخ به درخواست در postman قابل مشاهده است.



شکل ۴: خروجی براساس prompt داده شده

۵ نتیجه گیری

در این گزارش، نحوه استفاده از Spring AI برای یکپارچه سازی قابلیت های هوش مصنوعی در برنامه های وب بررسی شد. همان طور که مشاهده شد، این فریمورک امکان ارسال درخواست های هوش مصنوعی به مدل های مختلف را فراهم کرده و در این مثال از Stability AI برای تولید تصویر از متن استفاده شد.

فرایند پیاده سازی شامل ایجاد یک REST API برای دریافت متن از کاربر، ارسال آن به Stability AI، دریافت تصویر رمزگذاری شده و نمایش آن در خروجی بود. این روش نشان می دهد که با استفاده از Spring AI، می توان قابلیت های پیشرفته یادگیری عمیق را بدون نیاز به توسعه مدل های پیچیده به

برنامه‌های وب افزود.

با توجه به نتایج این تحقیق، می‌توان از Spring AI در سناریوهای مختلف مانند پردازش متن، ترجمه، پردازش تصویر و سایر کاربردهای هوش مصنوعی در برنامه‌های وب استفاده کرد. این رویکرد نه تنها توسعه سریع‌تر را امکان‌پذیر می‌کند، بلکه امکان بهره‌گیری از قدرت مدل‌های پیشرفته را نیز فراهم می‌آورد.

مراجع

<https://spring.ai/docs> [۱]

<https://huggingface.co/docs/api-inference/en/index> [۲]

<https://platform.openai.com/docs/guides/images> at: Available .OpenAI DALL · E 2 Documentation [۳]

<https://stablediffusionweb.com/api> [۴]

<https://deepfloyd.ai> [۵]

.