

2023

GPT结合工程化实现智能研发提效

主讲人：第一名的小蝌蚪

前言

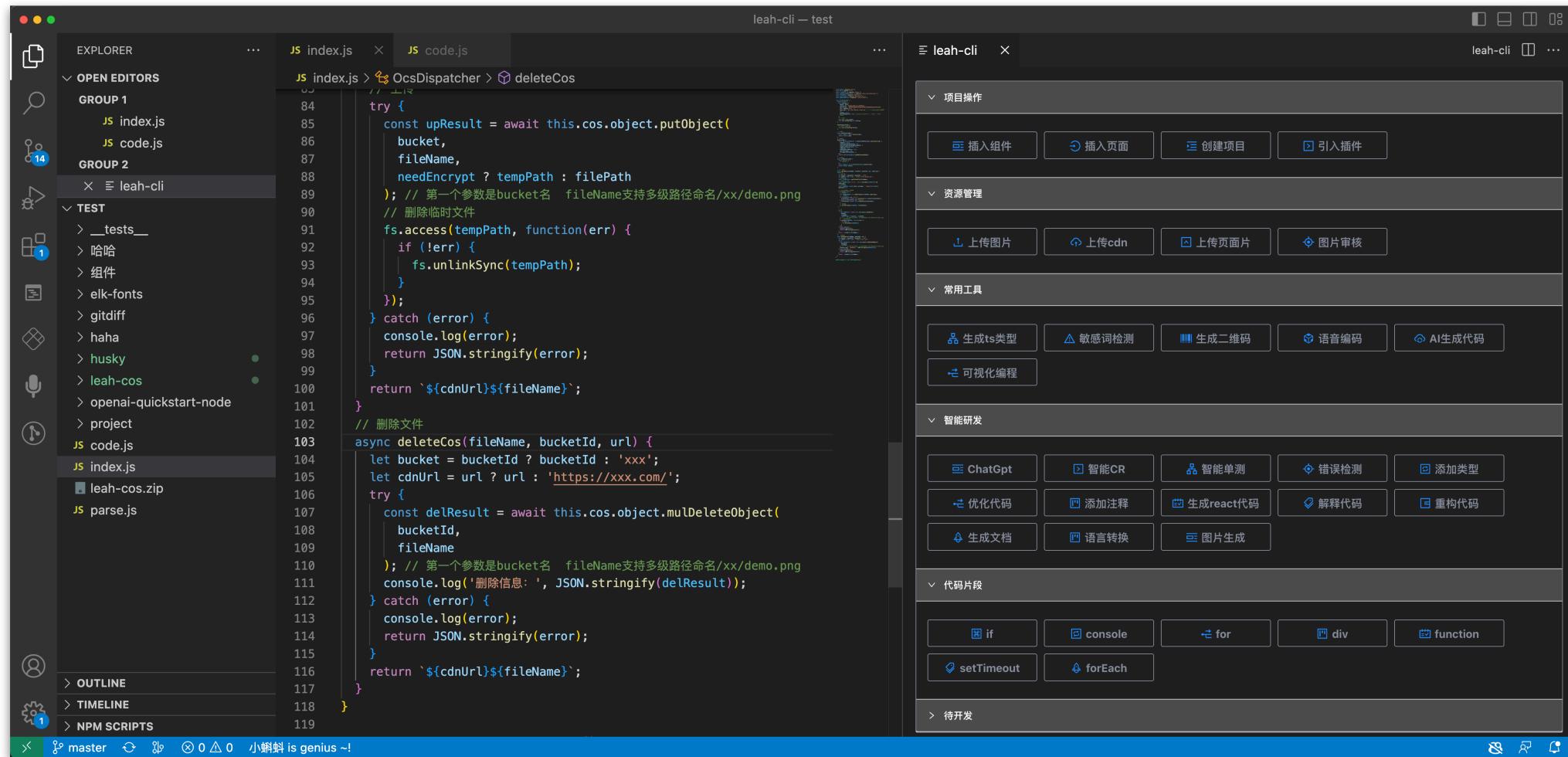
- 各大公司现状，降本增效，AIGC、审核、问答、研发
- 心态的变化：焦虑 -> 就这? -> 认可
- prompt已被列为公司资产，等效于代码
- 会共存，不会完全取代，拥抱变化高端稀缺、低端取代

A large, abstract graphic element consisting of several overlapping blue circles of varying sizes, centered on the left side of the slide.

PART 01

研发提效

智能工程化 - IDE插件 + AI



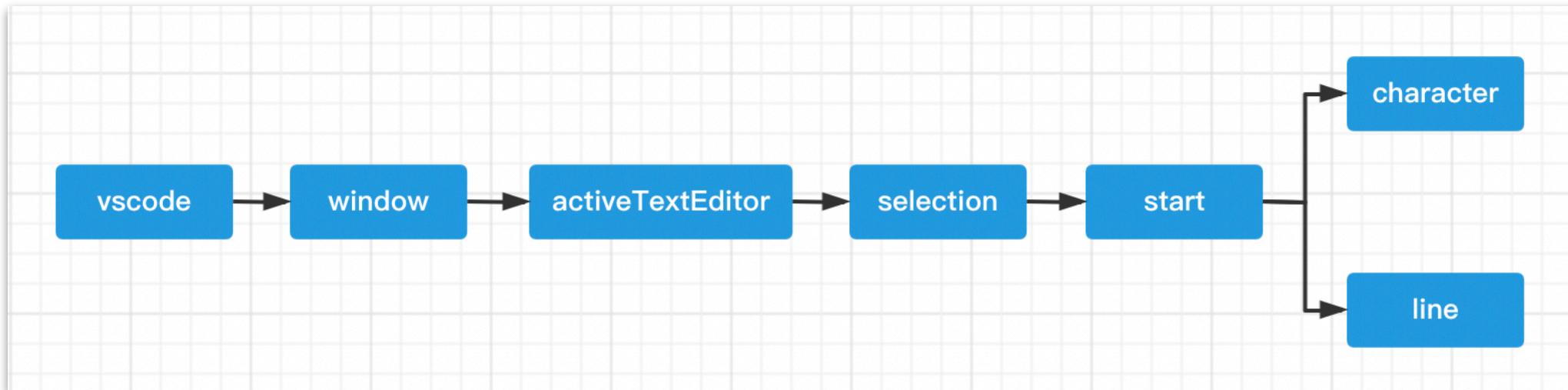
- 代码: <https://github.com/airuikun/smart-ide>

打破传统cli无法获取代码上下文的限制

```
leah > src > services > js todos.js > [e] fetch
1 import { http } from '@utils'
2
3 export const fetch = async () => await http({
4   method: 'get',
5   url: '/api/todos/fetch',
6 })  
└─▶
```



▼ start:
character: 2
line: 5



获取到用户鼠标锚点在代码中的具体行号和字符数

通过AST精准修改代码



1. **AST**: 通过parser将代码转化成AST抽象语法树
2. **定位**: 通过travse遍历AST找到代码需要插入的结构位置
3. **生成代码**: 通过Declaration和Specifier生成对应的代码语法树，Generator生成对应的代码

将重复枯燥的事情进行自动化、工程化



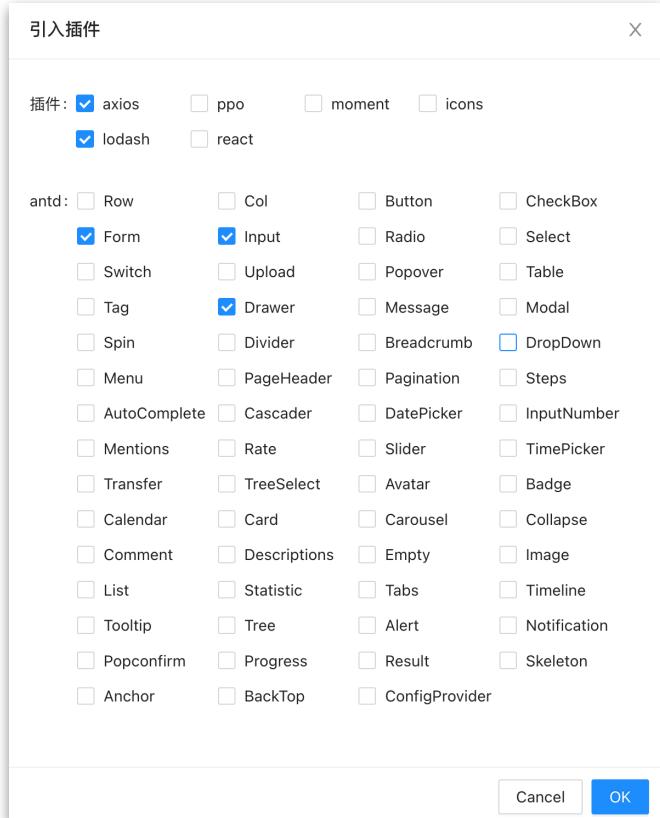
插入预设组件

```
import logo from './logo.svg';
import './App.css';
import ABC from './ABC';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <h1>ABC</h1>
      </header>
    </div>
  );
}

export default App;
```

将重复枯燥的事情进行自动化、工程化



插入预设代码

```
import axios from 'axios';
import lodash from 'lodash';
import { Form, Input, Drawer } from 'antd';
```

虽然无数套lowcode平台，但依然需要写基础代码

结合chatgpt

代码优化：

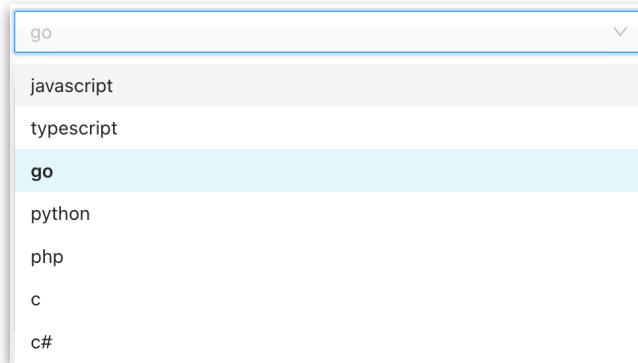
- 智能CR
- 智能单测
- 错误检测
- 解释代码
- 重构代码
- 优化代码

代码生成：

- 添加类型
- 添加注释
- 生成代码
- 语言转化
- 生成文档
- 图片生成

代码治理、项目迁移、扩展研发能力范围

语言转化



转化



A screenshot of a developer tool's interface. In the center, a modal window titled "语言转换" (Language Conversion) is open. The input field contains "go". Below it, the original Go code is shown:

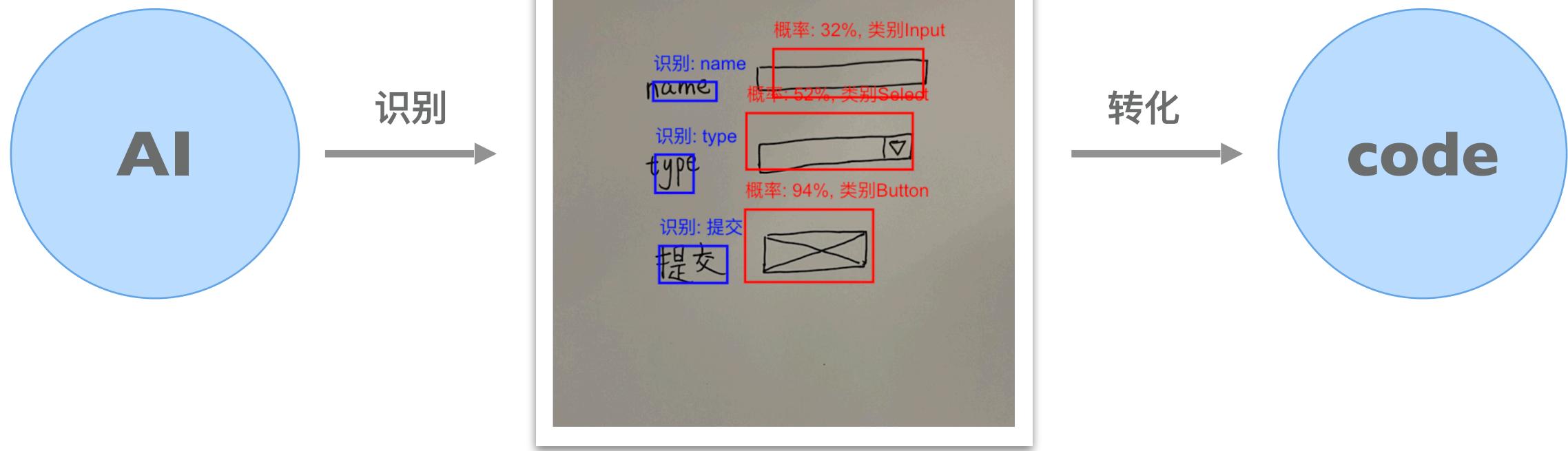
```
async deleteCos(fileName, bucketId, url) {  
    let bucket = bucketId ? bucketId : 'xxx';  
    let cdnUrl = url ? url : 'https://xxx.com/';  
    try {  
        const delResult = await this.cos.object.mulDeleteObject(  
            bucketId,  
            fileName  
        ); // 第一个参数是bucket名 fileName支持多级路径名  
    } catch (err) {  
        console.error(err);  
    }  
}
```

Below the original code, the converted JavaScript code is displayed, enclosed in a red rectangular box:

```
func deleteCos(fileName string, bucketId string, url string) string {  
    var bucket string  
    if bucketId != "" {  
        bucket = bucketId  
    } else {  
        bucket = "xxx"  
    }  
    var cdnUrl string  
}
```

At the bottom right of the modal, there is a blue "完成" (Finish) button.

手画稿生成代码

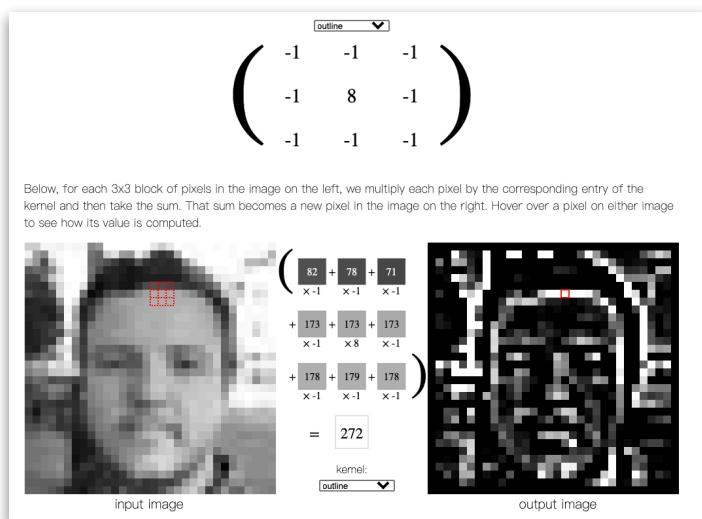
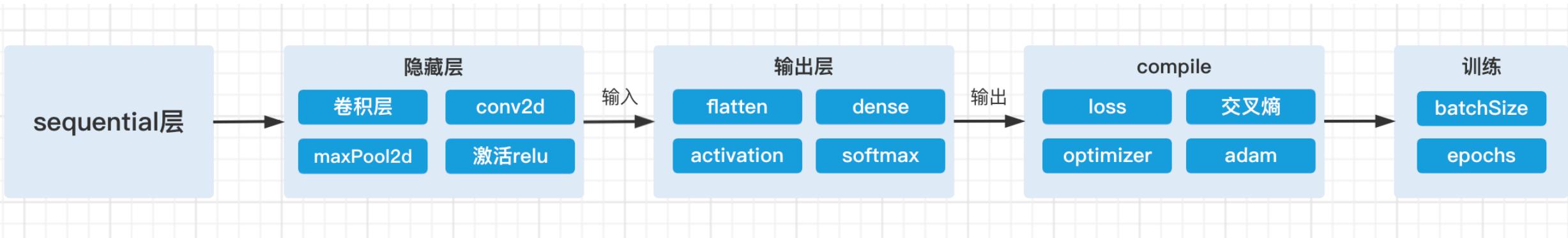


A screenshot of a Mac OS X desktop environment. The main window is a dark-themed code editor, likely Visual Studio Code, displaying a file named `App.js`. The code is a simple React component definition:

```
my-app > src > App.js > App
1 import logo from './Logo.svg';
2 import './App.css';
3
4 function App() {
5   return [
6     <div className="App">
7       <header className="App-header">
8         <img src={logo} className="App-logo" alt="Logo" />
9         <p>
10           Edit <code>src/App.js</code> and save to reload.
11         </p>
12         <a
13           className="App-link"
14           href="https://reactjs.org"
15           target="_blank"
16           rel="noopener noreferrer"
17         >
18           Learn React
19         </a>
20       </header>
21     </div>
22   ];
23 }
24
25 export default App;
26
27
```

The code editor has several status icons on the left: a file icon with 6, a search icon with 1, a refresh icon with 6, and a gear icon. The top bar shows the title "App" and the status bar indicates "156字". The bottom dock contains various application icons, including Finder, Mail, Calendar, Safari, and others.

模型训练

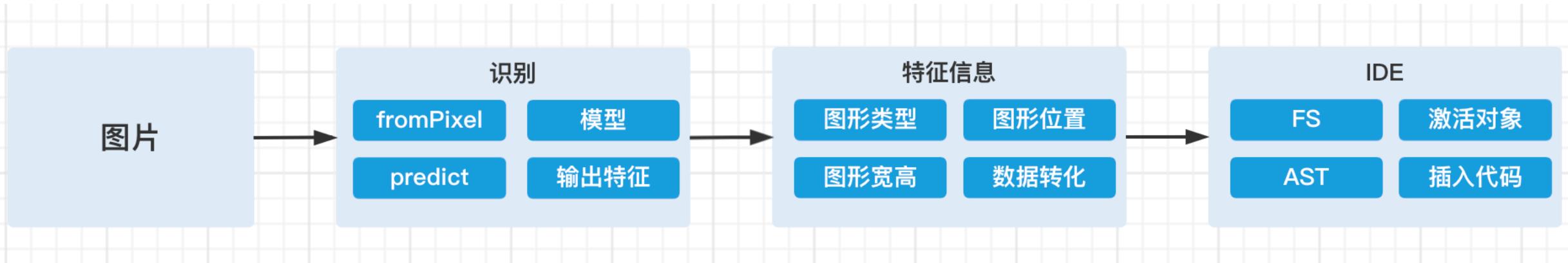


构建: 定义model层，设定隐藏层激活函数为relu，设定卷积层、卷积核与池化层

编译: 编译阶段设定损失函数为交叉熵，优化器为adam

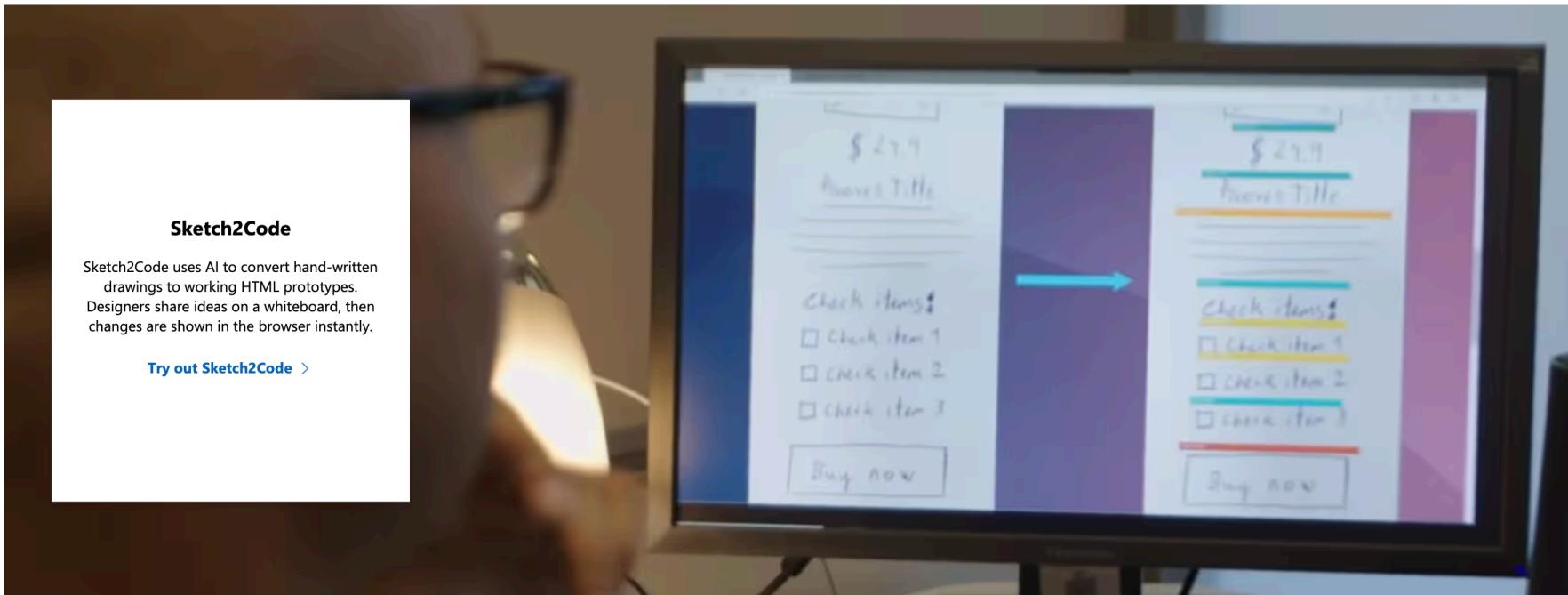
训练: 训练阶段设定对应的batchSize和epochs

图形识别并生成代码



1. **识别:** 通过fromPixel获取到图片数据，喂给模型进行预测，输出图形识别结果
2. **特征:** 识别出的数据包括图形的类型、位置、宽高等信息，然后转成前端可识别的json数据
3. **代码:** IDE拿到结果数据，将对应的模板代码文件拷贝，并获取位置信息，通过AST操作插入代码

微软sketch2code



让AI帮我们把前30%的事情做完

A large, semi-transparent graphic element consisting of three concentric circles in varying shades of blue, centered on the left side of the slide.

PART 02

未来畅想

趋势

- IDE插件化、全栈化、原子化、Lowcode+AI
- prompt工程师 = 超级个体 = 研发 + 测试 + 产品 + UI + 客服
- 认知变现：在未来最值钱的是想法idea

谢谢

- 第一名的小蝌蚪
- 代码：<https://github.com/airuikun>