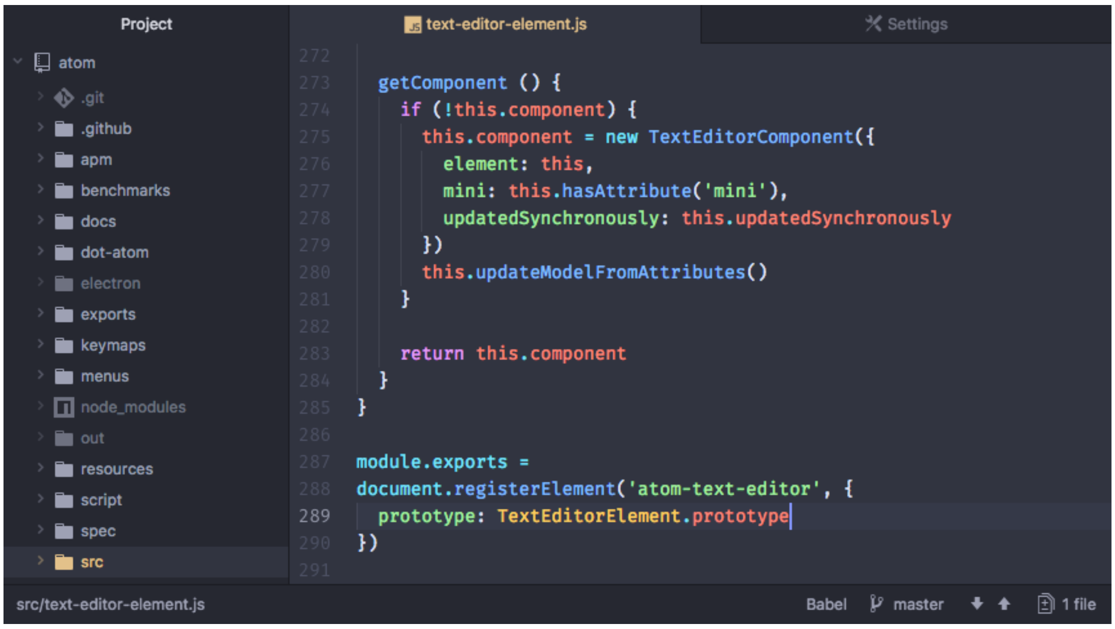
# Software and Software Engineering

## Software Applications:

### 1- System software

System software is a collection of programs written to service other programs, (e.g., compilers, editors (Figure 2.1), file management utilities (Figure 2.2), operating system components, networking software, and drivers). The system software area is characterized by heavy interaction with computer hardware; heavy usage by multiple users; concurrent operation that requires scheduling, resource sharing, and sophisticated process management; complex data structures; and multiple external interfaces.



```

  getComponent () {
    if (!this.component) {
      this.component = new TextEditorComponent({
        element: this,
        mini: this.hasAttribute('mini'),
        updatedSynchronously: this.updatedSynchronously
      })
      this.updateModelFromAttributes()
    }

    return this.component
  }
}

module.exports =
document.registerElement('atom-text-editor', {
  prototype: TextEditorElement.prototype
})

```

Figure 2.1: Editor software (Atom).

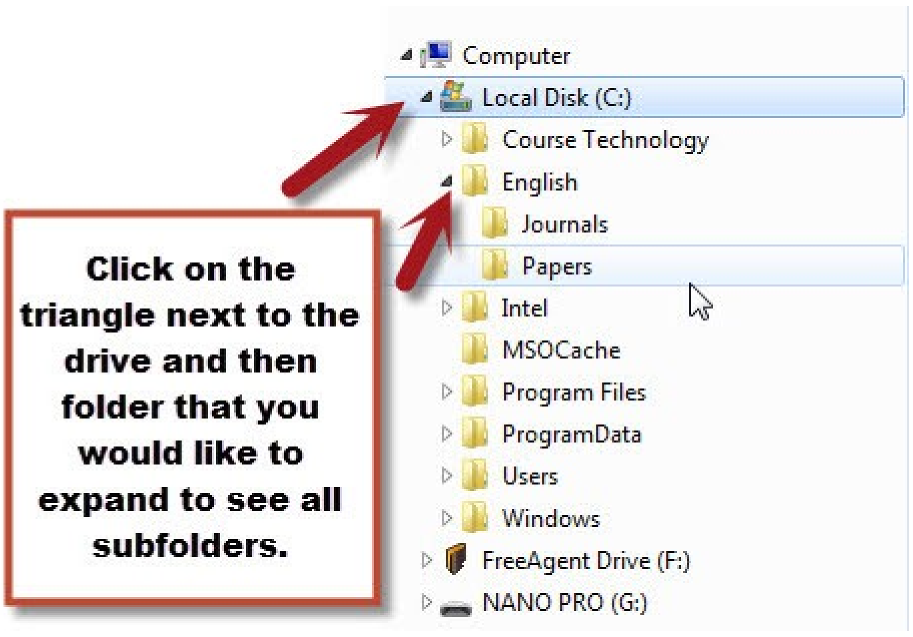


Figure 2.2: File management utilities (File Explorer).

## 2- Real-time software

Software that monitors/analyzes/controls real- world events as they occur is called ***real time.*** Elements of real-time software include a data gathering component that collects and formats information from an external environment, an analysis component that transforms information as required by the application, a control/output component that responds to the external environment, and a monitoring component that coordinates all other components so that real-time response (typically ranging from 1 millisecond to 1 second) can be maintained. For example, air traffic control system and railway switching system.

## 3- Business software

Business information processing is the largest single software application area, (e.g., payroll, accounts receivable /payable, inventory) have evolved into management information system (MIS) software that accesses one or more large databases containing business information. Applications in this area restructure existing data in a way that facilitates business operations or management decision making.

## 4- Engineering and scientific software

Modern applications within the engineering/scientific area are moving away from conventional algorithms. For example, Computer-aided design, system simulation, and other applications.

## 5- Embedded software

Resides in read-only memory and is used to control products and systems for the consumer and industrial markets, (e.g., keypad control for a microwave oven, digital functions in an automobile such as fuel control, dashboard displays, and braking systems).

## 6- Personal computer software

Word processing, spreadsheets, computer graphics, multimedia, entertainment, database management, personal and business financial applications, external network, and database access are only a few of hundreds of applications.

## 7- Web-based software

The Web pages retrieved by a browser are software that includes executable instruction (e.g., CGI, HTML, Perl, or Java), and data (e.g., hypertext and a variety of visual and audio formats), in essence, the network becomes a massive computer providing an almost unlimited software resource that can be accessed by anyone with a modem.

## 8- Artificial intelligence software

Artificial intelligence (A1) software makes use of non-numeric algorithms to solve complex problems that are not amenable to computation or straightforward analysis. Expert systems, also called knowledge-based systems, pattern recognition (image and voice), artificial neural networks, theorem proving, and game playing.

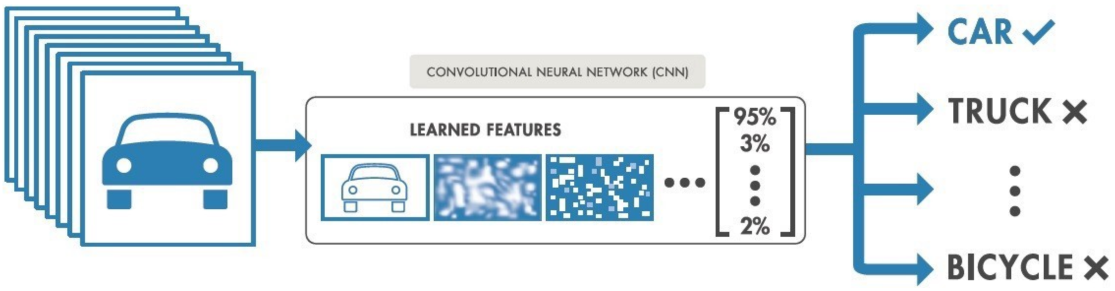


Figure 2.3: pattern recognition (image classification).

## Software Engineering Goals:

- Efficiency—The software is produced in the expected time and within the limits of the available resources. The software that is produced runs within the time expected for various computations to be completed.

- Reliability—The software performs as expected. In multiuser systems, the system performs its functions even with other loads on the system.

- Usability—The software can be used properly. This generally refers to the ease of use of the user interface but also concerns the applicability of the software to both the computer's operating system and the application environment.

- Modifiability—The software can be easily changed if the requirements of the system change.

- Portability—The software system can be ported to other computers or systems without major rewriting of the software.

- Testability—The software can be easily tested. This generally means that the software is written in a modular manner.

- Reusability—Some or all of the software can be used again in other projects. This means that the software is modular, that each individual software module has a well-defined interface, and that each individual module has a clearly defined outcome from its execution.

- Maintainability—The software can be easily understood and changed over time if problems occur. This term is often used to describe the lifetime of long-lived systems such as the air traffic control system that must operate for decades.

- Correctness—The program produces the correct output.

## The Process Framework

A process framework establishes the foundation for a complete software engineering process by identifying a small number of framework activities that are applicable to all software projects. A generic process framework for software engineering encompasses five activities:

- Communication. Before any technical work can start, it is critically important to communicate with the customer. The intent is to gather requirements that help define software features and functions.

- Planning. A software project is a complicated journey, and the planning activity creates a "map" that helps guide the team as it makes the journey. The map (called a software project plan) defines the software engineering work by describing the technical tasks to be conducted, the risks that are likely, the resources that will be required, the work products to be produced, and a work schedule.

- Modeling. A software engineer creates models to better understand software requirements and the design that will achieve those requirements.

- Construction. This activity combines code generation and the testing that is required to uncover errors in the code.

- Deployment. The software is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation.

The work associated with software engineering can be categorized into generic phases:

1-Requirements gathering and analysis.

2-Design.

3-Coding.

4-Testing.

5-Maintenance.