



(RESEARCH ARTICLE)



## Vision-guided automation: A generic approach to web form filling using GPT and computer vision

Leela Gowtham Yanamaddi <sup>1,\*</sup> and Balaji Kummari <sup>2</sup>

<sup>1</sup> CEO and VP of Engineering, scale.jobs 537 Payne Rd, Woodstock, GA, USA 30188.

<sup>2</sup> CTO, scale.jobs 1-84, Beside Venugopala Swamy Temple, Rayanapadu, Vijayawada, AP 521241, India.

World Journal of Advanced Research and Reviews, 2023, 20(03), 2096-2107

Publication history: Received on 03 November 2023; revised on 16 December 2023; accepted on 18 December 2023

Article DOI: <https://doi.org/10.30574/wjarr.2023.20.3.2524>

### Abstract

Combining computer vision approaches with GPT (Generative Pre-trained Transformer) models, this research presents a novel approach to automating web-based form filling tasks. A general approach that can adapt to different forms without knowing their structure is made possible by the suggested system, which detects and labels interactive elements on web pages visually. This allows it to transcend the restrictions of hardcoded DOM element interactions. Notable advancements include utilising computer vision to recognise and label form elements and integrating GPT models to read form fields semantically and produce context-appropriate responses (for instance, using resume data). Plus, AI-guided judgements are made using a versatile action system that mimics human-like interactions like typing, clicking, and scrolling. An automated job application form filling case study demonstrates the system's efficacy and highlights its potential for wide-ranging online automation activities.

**Keywords:** Vision-Guided Automation; Web Form Filling; GPT; Computer Vision

### 1. Introduction

Technological developments in automation have recently ushered in a new era of enhanced company efficiency and output. One of them, RPA for Chrome, takes automation to a whole new level by integrating computer vision with Automat AI's GPT-4. This makes automating tasks much simpler. Understanding the basics of this cutting-edge technology and how it might affect companies is the focus of this essay.

#### 1.1. What is Automat AI's GPT-4?

Set up OpenAI's GPT-4 is a model for next-gen natural language processing. When it comes to creating conversation, questions, and content that seems natural, GPT-4 really shines. This approach completely changes the game when it comes to task automation by giving it the ability to understand and generate text-based information.

#### 1.2. The Role of Computer Vision

The field of computer vision is dedicated to the analysis and interpretation of visual data. For jobs like object detection, captioning, and pattern recognition, computer vision examines visual data from pictures, videos, and websites. In RPA, visual information processing and comprehension are made possible only using this technology.

\* Corresponding author: Leela Gowtham Yanamaddi

### **1.3. What is RPA for Chrome?**

With RPA for Chrome, you can automate your online browsing and other web-based operations. Information collection, data entry, form filling, online scraping, and other web-based operations can be automated with this approach. Powerful automation is possible with RPA for Chrome and GPT-4 when you add computer vision to the mix.

### **1.4. Collaboration of GPT-4 and Computer Vision**

By working together, GPT-4 and computer vision are able to complete tasks such as:

#### **1.5. Information Gathering**

In order to retrieve relevant data from websites, GPT-4 adheres to instructions given in spoken language, whereas computer vision gathers and interprets data from images.

#### **1.6. Data Processing**

GPT-4 deciphers the data and gives the appropriate commands. Visual input is processed by computer vision systems to carry out activities such as object detection and pattern recognition.

#### **1.7. Task Automation**

Automating tasks and controlling computer vision are also responsibilities of GPT-4. Automated web-based tasks can be carried out by means of RPA scripts.

#### **1.8. Potential Impact on Businesses**

Companies may feel a variety of effects from this integrated approach:

##### **1.9. Improved Productivity**

Task automation improves efficiency by facilitating the rapid and precise completion of routine operations and data collection.

##### **1.10. Increased Accuracy**

Computer vision and GPT-4's natural language comprehension both contribute to better quality and less mistakes when interpreting visual data.

##### **1.11. Scalability**

This method can handle a lot of data and jobs, so it's flexible enough to handle when your organisation grows.

##### **1.12. Strategic Insights**

Automation frees up workers to focus on higher-level, more strategic duties, which in turn helps them to gain valuable insights on those jobs.

The complex network of code that powers our online interactions often hides threats in today's dynamic software development environment. Security flaws in online apps are growing in number and sophistication in tandem. It can be challenging to ensure data security and implement appropriate access control with new development paradigms brought about by front-end frameworks like Angular [1]. This is particularly true when automation is needed, which reduces human contact. This is the backdrop against which large language models (LLMs) like ChatGPT [3], the different GPT models [2], have emerged, with the promise to transform tasks like code generation, documentation, and debugging. There is still a lot of discussion and research going on regarding their actual capabilities for understanding, assessing, and enhancing code. The issue of whether these models can understand code complexities to find security flaws and vulnerabilities is particularly important. This is especially important in the context of today's intricate web frameworks, where the loss of sensitive information could occur from using the wrong solutions.

##### **1.13. AI in Software Development**

Our primary goal is to use what we know about the changing software development landscape to investigate the role of AI, and more specifically big language models, in this area. The development of LLMs has been picking up speed recently, and in the last year, thanks to services like ChatGPT [3] and others like it, their popularity and use have reached new

heights. Researchers from all over the world are delving into the potential benefits and drawbacks of these models for a variety of industries, including but not limited to the information technology (IT) sector, programming, education, healthcare, administration, and business, as well as how they can optimise processes and enhance decision support. The precise limitations of the models in terms of understanding and interpretation are still not evident, even if there is increased focus on the optimality and quality of the code created by generative pre-trained transformers (GPTs). A comparable difficulty arises when thinking about interpretability, or how well a big language model understands the code's purpose and function so that it can test, debug, or improve it with little to no input from the developer. This type of code interpretation and debugging assisted by AI is quite interesting. These systems, armed with the correct semantic and logical understanding, can probably find places where bugs in the code could be hiding, places that would otherwise require a lot of testing to find. In a separate study, we used GPT to help with static code analysis of Angular applications, and we found that it outperformed BERT (which relies solely on prompting and doesn't require any additional model training) on the problem set we were looking at [4]. Recent tendencies indicate that testing is one area where AI-enabled technologies are gaining popularity [5]. For big, complicated code bases, the importance of aggregating and making the code base more transparent and manageable to successfully manage configurations like security configuration enforcement and vulnerability analysis is not new. In an effort to facilitate clearer and easier software system security setups, Heydon et al. [6] presented Miró in 1990 as a collection of visual description languages. The Vicoms framework, introduced by Giordano and Polese [7], facilitated the administration of role-based access control settings for Java programs in a way that was separate from the source code but yet easily understood, serving a similar purpose. The aforementioned code interpretation is always going to have an impact on the testing, usage, and implementation of such a sophisticated tool.

#### **1.14. The Challenges of Contemporary Web Frameworks**

After going over the possibilities and strengths of LLMs in software development, we need to dive into the intricacies of modern web frameworks. As an example, Angular offers a paradigm that is powerful but has its own challenges. In order to properly employ LLMs to resolve these complications, it is essential to understand them. New difficulties arise as a result of the design of contemporary web frameworks, such as Angular [1]. For example, because of modular nesting and partial component navigation, it becomes challenging to construct a DOM structure that matches the call network and dependency graph of single-page applications. Analytically speaking, even the original Angular was difficult. One such tool that Misu et al. [8] developed to help developers find anomalies in static source code is FANTASIA, which analyses AngularJS MVCs using an AST-based approach. Debugging and design have both benefited from TypeScript's use of object-oriented paradigms, typology, and strong modularity. As with other modern front-end frameworks, static and manual analysis have been hindered by the increased modularity, extensive call and dependency trees, and complicated navigation. While developing these solutions, our research team looked into data security and access management [9], with a focus on improving the implementation of access control, since we had previously worked on a number of healthcare applications. The primary goal was to quickly and efficiently retrieve the necessary data from the databases in a way that complied with ethical standards and allowed only authorised organisations access to the data. But there were restrictions at the front-end applications' data path ends; evaluating the security of the web apps or applications there would necessitate quantifying their security, and this data is essential for evaluating the data path's overall security. Although this paper primarily focusses on Angular, the methods it presents should be applicable to other frameworks that face comparable challenges. The core ideas of our method, which include creating a code base map, carrying out the vulnerability identification procedures, and combining incomplete results, are independent of the programming language. They are flexible enough to function with a wide range of front-and back-end frameworks. One standardised catalogue of possible vulnerabilities in software and hardware design is the common weakness enumeration (CWE) [10], which is put together by the MITRE Corporation. A consistent strategy for detecting and mitigating these vulnerabilities is provided by this complete framework to the technological world. Automated tools are good at finding many vulnerabilities fast when they follow CWE's instructions. Still, automated identification isn't always possible due to flaws' subtleties; this is especially true when the vulnerabilities are deeply embedded in the code's semantics. It takes more than just analysing code to understand the vulnerabilities in some apps. Data type, origin, flow, and destination can all impact possible weak spots in a processing system. Consideration of the regulatory and ethical implications of data management may reveal vulnerabilities in an application that deals with sensitive medical records, for example [11]. The same holds true for culturally or politically specific contexts; apps running in such an environment may be susceptible to different types of attacks. Because experts not only decipher code but also understand the bigger picture of an application's environment, this emphasises the paramount need of human competence. The interplay between an app's environment, the data it handles, and its source code is getting more intricate as cyber threats change and adapt. The best approach to system security, which is the zenith of vulnerability detection and treatment, is to combine automated techniques with the deep contextual understanding of human specialists.

## 2. Literature review

The CWE-653 vulnerability, which stands for "improper isolation of compartmentalisation," is widely present in applications that are used for the front end. A vital process or action is implemented in the code without sufficient separation, which leaves web applications vulnerable. Therefore, it is risky for the same piece of code to handle both important and less important data. This could lead to data leakage or unauthorised use if developers aren't paying attention or if malicious users are using the code. Examples of CWE members that are commonly found in front-end applications include CWE-203, which means "observable discrepancy," CWE-359, which means "exposure of private personal information to an unauthorised actor," and CWE-497, which means "exposure of sensitive system information to an unauthorised control sphere." Identifying sensitive data was the first step in our strategy. Code should prioritise isolating, protecting, and controlling access to sensitive data because its misuse might have serious consequences. Protecting the entities that handle these operations is crucial for progressive web applications. In the Angular framework, this means ensuring the service classes, which often use the classic data object access (DAO) pattern, are well-protected. This is because we assume that this data comes from a server or database connection and is passed to the web application. The Angular framework allows these services to function as singletons, and the classes responsible for the user interface elements or other services can access and invoke them through injection. The research questions posed in this paper originated from the need to determine if the component classes calling these services have proper access control. In the Angular framework, this is achieved through the AuthGuards, which are classes that implement the CanActivate interface. When attached to the application's navigation rules, these classes can block or redirect users from accessing user interfaces that they do not have authorisation to see.

### 2.1. Sensitivity and Potential Vulnerabilities

Data sensitivity and possible vulnerabilities are the next important topics to cover after modern web frameworks. Protecting private information is a top priority in the world of web apps, particularly those built with Angular and similar frameworks. The development of secure and robust software relies on accurately identifying different levels of sensitivity and the possible consequences of data leaking. One major roadblock we hit was the following: Is it feasible to categorise and quantify the sensitivity of sensitive data? Without a doubt, a whole new set of worries about understanding, handling, and expecting the handling of personal data has been generated by the technological advancements of the last decade, in addition to the numerous scandalous breaches of personal data by big tech companies against their users. We comprehend sensitive data in different fields [12], how personal sensitive data came to be, and to what extent it varies [13], among other topics that have been the subject of a great deal of research in recent years.

A number of recent trends have emerged in the field of data science, including data collection parsimony, the measurement of how little data machine learning algorithms need for efficient learning and operation [14], the classification and quantification of data sensitivity [15,16], and the automatic identification of sensitive data [17]. We classified sensitive information into three groups according to the harm that could result from its exposure; the methodology section goes into additional detail on this topic. On one end of the spectrum, we have data that is very crucial, and on the other, we have data that is just potentially harmful due to the buildup of a huge amount of leaked data. While each development team may put their own spin on classifying such a set of methods, our goal was to provide an easily-defined, chain-of-thought-descriptible scale for big language models.

Presented below are the main findings drawn from this study:

- Using the GPT API to define a custom classification and test it on a test set of 200 variable names (to measure data sensitivity and immunity). A solid foundation for understanding and categorising data according to its sensitivity is provided by this classification. To make our method more practical, we test it with the GPT API to make sure the model can correctly identify and handle different types of data.
- A complicated web application's code can be statically analysed by showing how prompt engineering for the GPT API is used to build prompts that consider both the surface and deeper contexts. By quickly engineering, we make sure that the static analysis is accurate and takes context into account, allowing the GPT models to reach their full potential. This method provides comprehensive and in-depth insights by combining generic AI analysis with vulnerability-specific software testing.
- A comparison of the GPT-3.5 and GPT-4 models' abilities to use static code analysis to find sensitive data in an application. We hope to be able to tell how the GPT models have changed and improved over time by comparing their performance. Results from this analysis show that improvements to software application security can be achieved through the use of LLMs.

- Assessing the GPT-3.5 and GPT-4 models' capacities to ascertain the degree of protection for components of front-end applications. Ensuring proper protection of sensitive data is just as important as detecting it. To gain a better understanding of the GPT models' possible function in security evaluations and interventions, we can test their capacity to predict protection levels.
- Assessing how well the GPT-3.5 and GPT-4 models identify potential vulnerabilities caused by inadequate isolation and protection of sensitive data handling in online applications. To understand the real-world use of LLMs in cybersecurity, this assessment is crucial. These models have the potential to become vital for app developers looking to protect their apps from attacks if they can accurately detect flaws in data management.

---

### **3. How to incorporate GPT models: testing 6 CHATGPT use cases**

#### **3.1. The task is viable by means of the existing model.**

In this case, the model serves as a smart catalogue, leveraging GPT's reaction to speed the user's content browsing and decision-making processes. Unfortunately, there is no option for customisation or personalisation.

#### **3.2. Fine-tuning the model with your own data.**

In a variety of contexts, such models can be helpful for developing a user-centric recommendation system for chatbots. You can train the GPT model to incorporate personalisation using your own data. Several criteria determine how much data is required to fine-tune a GPT model. These include the dataset's size and complexity as well as the intended model performance. For a GPT model to be fine-tuned, it usually takes several hundred cases. On the other hand, getting a model to perform at its best in complicated domains can require thousands—if not millions—of samples. I suggest going through these 5 stages to figure out if the model needs tweaking:

#### **3.3. Business understanding**

It necessitates familiarity with the business case, the model's assigned tasks, the data it should utilise, and the anticipated personalisation features.

#### **3.4. Data gathering**

We can skip this step if we can omit providing personalised information in the beginning. If you intend to apply the GPT model to a very particular domain, however, it will malfunction by design. Accordingly, this step is vital. Providing the user with clothing recommendations that take into account the store's inventory is one example. Subsequently, you'll need to gather details about the store's products and offer some basic guidelines for making recommendations.

#### **3.5. Modeling**

Lastly, producing the model and putting those modifications into action. Remember that the GPT model does not come pre-packaged with all the domain areas covered. Additionally, you will require the application and backend management tools for it. Next, natural language processing (NLP) algorithms need to be applied to the GPT model's output for any significant information extraction to take place. The model provides a response in the form of a list of sentences. The following steps of the pipeline will use this output as a filter or keywords, so it's important that it be supplied as one or more words. As a result, additional models will be necessary for the GPT model to function.

#### **3.6. ChatGPT Business Use Cases**

By utilising the RLHF approach for its training, ChatGPT has significantly improved over its predecessors. When contrasted with the equally well-liked GPT 3 model, this yielded noticeably better results. As a result, ChatGPT's natural language processing (NLP) features are being used for a variety of business and technical purposes. How does ChatGPT feel about its potential business applications? The model's response, which includes a list of typical use cases, is shown here. We may outsource a plethora of low-level activities to ChatGPT because the model is dependent on data that is only available up to the year 2021. Additionally, this is by no means an exhaustive list. Other options include concentrating on a single domain area and seeing whether we can get any value out of it: Many inferences about three highly promising business use cases may be drawn from these examples:

### **3.7. Customer support**

With your own data, you can make ChatGPT a highly capable conversational bot. It is a ready-made answer for customer service systems, especially chatbots, since it can understand the question correctly even with incomplete or incorrect grammar or syntax.

### **3.8. Lead generation**

Continuing from where we left off, lead generating operations can make advantage of these same capabilities, as the ChatGPT level of speech is often not too far from human writing ability.

### **3.9. Data analytics**

Modern BI and data analytics solutions may face competition from GPT approaches in the future. Reason being, they can impart useful insights to human workers while processing massive amounts of historical data. Nevertheless, GPT models seem to be awkward when it comes to analysing data in real-time, which necessitates continual updates and the most current information accessible. In addition to the complexity of the RLHF training process, a machine learning pipeline including automated retraining and model redeployment is required. This is, at least in the short term, somewhat complex.

Consequently, GPT models are still not ready for use in applications that require constant data updates in real time. What other uses does it have?

---

## **4. Copywriting with ChatGPT**

A widely held belief regarding language models and neural networks is that they have the potential to significantly increase writing speed or even replace copywriters entirely. Today, we'll find out if it's true. To test the model's ability to write articles about software engineering, we will utilise two different kinds of requests. I want the model to describe things and provide an expert opinion or reasons for the second type of request, while the first type just presents the text's main notion. From an information aspect, the model responds adequately to both scenarios. Texts are also excellent in terms of style and grammar. On the other hand, it appears quite ordinary and unconvincing in terms of originality, readability, logic, and the defence of a certain position. Although ChatGPT can ease your marketing efforts with its blueprints, human proofreaders face one drawback with it. It seems challenging to locate the information source and validate the model's assertions because the model analyses the data independently. For ChatGPT to be truly effective in marketing, it needs to be fine-tuned and equipped with extra capabilities like AI explainability.

---

## **5. General Programming with ChatGPT**

The ability to use programming languages to create applications is a natural extension of GPT models' ability to communicate in human languages. Anton Logvinenko, who heads our PHP/DevOps division, tested this to see what happens when we use the model to write PHP code for different functions.

---

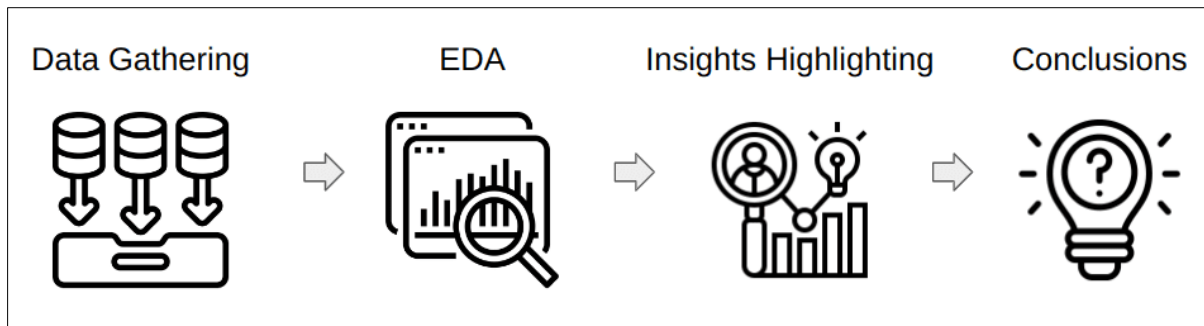
## **6. Building AI pipelines with the help of GPT Models**

Since GPT models are capable of handling most natural language processing tasks, they might be seen as a substitute for training your own language models. In keeping with this practice, I will lay out the groundwork by challenging ChatGPT to explain how it may outperform training a custom NLP model: The shortcomings of this strategy must be recognised. You can only use GPT models to launch services without collecting initial data because they do not have access to real-time data. Although this isn't a quick remedy for all the particular NLP tasks, you will eventually be able to gather live data from the model's interactions with users. But there are plenty of other data science applications where GPT models could be useful.

---

## **7. Processing Table Data with ChatGPT**

The processes of exploratory data analysis (EDA) are necessary for the construction of models for categorisation, regression, and forecasting. With the help of EDA, we may discover the connections between various data aspects, identify insights and outliers, and figure out why some unusual behaviour occurred. Consequently, there are often four stages to data science projects:



**Figure 1** EDA Pipeline

Reducing the aforementioned process to a single well-formulated request using the GPT model is possible. Take this example: we want ChatGPT to draw conclusions based on Amazon sales data broken down by quarter.

### 7.1. ChatGPT for recommendation systems

On many different platforms and in many different kinds of applications, recommendation systems, often called recommender engines, are now as ubiquitous as search engines. An essential function of recommendations is to learn each user's tastes so they can provide personalised content suggestions that will entice them to stay on the site for as long as feasible. However, not everyone can afford recommendations, even though they offer competitive advantages. This is likely due to the fact that building one requires collecting data, conducting experiments, and training a functional model. Before we can begin making recommendations, we often need to gather data on our material. Then, we need to track our clients' preferences for different types of content in a systematic way. Since ChatGPT already contains a plethora of data, it may be used instead of recommendation engines in this AI app development scenario. Although it lacks content-specific tuning and ignorance of user search histories, it is capable of providing somewhat targeted suggestions given sufficiently detailed input. Requests like "I want to have a journey" were commonplace. I enjoy spending time near the water and have a toddler. I don't know where to start walking. This is the result:

### 7.2. ChatGPT response

Travelling to a coastal area with a toddler in tow can be a wonderful experience. Some potential vacation spots that you and your loved ones might like are:

#### 7.2.1. The beaches of California

Sun, sand, and water are the perfect trio at California's beaches. Santa Monica, Newport Beach, and Huntington Beach are among the most visited locations.

#### 7.2.2. The beaches of Florida

Among the numerous stunning beaches in Florida are Clearwater Beach, Miami Beach, and Siesta Key.

#### 7.2.3. The beaches of Hawaii

Beaches in Hawaii are well-known for their powdery white sand and crystal-clear water. Poipu Beach, Kaanapali Beach, and Waikiki Beach are among the most visited locations in Hawaii.

#### 7.2.4. The beaches of the Caribbean

A few examples of the stunning beaches found in the Caribbean are Turks and Caicos' Grace Bay, Aruba's Eagle Beach, and Jamaica's Seven Mile Beach.

#### 7.2.5. The beaches of the Mediterranean

Spain, Greece, and Italy are just a few of the Mediterranean nations that boast stunning beaches. Positano, Italy; Mykonos, Greece; and Mallorca's Playa de Muro are among the most visited tourist spots in the world.

In addition to swimming, sunbathing, water sports, playgrounds, and theme parks, these locations also provide a wide array of additional activities and attractions that people of all ages can enjoy. Other considerations, such as accessibility,

weather, and safety, should be considered as well. What follows is an example of how the response can appear if we add details to our request:

### 7.3. Request

The desire to travel is strong within me. I have three thousand dollars, a two-year-old child, and I enjoy spending time near the water. "Where am I supposed to go?"

### 7.4. ChatGPT response

A vacation to the beach with a 2-year-old might be a wonderful plan. When making travel plans, it's wise to keep your budget in mind. If you're looking for affordable vacation spots for your family, here are a few ideas:

#### 7.4.1. *The beaches of Mexico*

Sun, beach, and sea are the perfect trio that Mexico has to offer. Cancun, Puerto Vallarta, and Playa del Carmen are among the most visited places in Mexico.

#### 7.4.2. *The beaches of Central America*

The beaches, climate, and prices in Nicaragua, Panama, and Costa Rica are unbeatable.

#### 7.4.3. *The beaches of Southeast Asia*

Beautiful beaches, pleasant climates, and low prices are what make Thailand, Vietnam, and the Philippines ideal vacation destinations.

#### 7.4.4. *The beaches of the Caribbean*

Cuba, Jamaica, and the Dominican Republic are a few Caribbean destinations that provide stunning beaches at affordable prices.

In addition to swimming, sunbathing, water sports, playgrounds, and theme parks, these locations also provide a wide array of additional activities and attractions that people of all ages can enjoy. Travelling with a 2-year-old also requires thinking about things like accessibility, safety, and the availability of high-quality lodging and amenities, such as decent childcare centres and hospitals. Attempts by the model to bypass the initial prompt result in output that isn't constrained by the parameters we supplied as requests. To acquire satisfactory results, it needs a human to utilise the proper terms, therefore even those raw results can perform better than trying to find content by applying dozens of filters. That is the main benefit of suggestions based on natural language processing compared to conventional methods.

### 7.5. Using ChatGPT for Computer Vision Tasks

We can't use GPT models to replace computer vision models because they are trained on text data. The following computer vision pipeline components, however, can be made easier to construct with ChatGPT:

- Image recognition based on colour information.
- Visual resemblance, such as icons that resemble McDonald's in terms of colour scheme or typeface.
- Detecting objects, such as those shown in images, logos, and other visual resources. OCR, which stands for optical character recognition.

Various domains can benefit from the data analysis and preparation made possible by ChatGPT for use in computer vision models. As an example, the model can be utilised to extract text from images in optical character recognition (OCR) jobs, such as ID card recognition. This is a different stage in preparing data for training.

---

## 8. AI decision-making

The field known as artificial intelligence (AI) focusses on programming computers to mimic human intelligence in terms of thought processes and behaviour. Visual perception, voice recognition, decision-making, and language translation are just a few of the activities that these machines are trained to execute. Artificial intelligence (AI) experts hold the belief that, given sufficient data, algorithms, and processing capacity, a machine can be trained to mimic human thought processes. In order for AI systems to learn to identify patterns, make predictions, or make judgements, they are trained



using massive volumes of data and algorithms. Artificial intelligence comes in several forms, such as: To put it simply, reactive machines are the most basic form of artificial intelligence; they can carry out a single task, but they can't remember anything or learn from their mistakes. This AI system has a limited memory, meaning it can learn from its mistakes but can't remember everything. It uses this knowledge to guide its future judgements. Artificial intelligence with a "theory of mind" can put itself in another person's shoes and react appropriately to their feelings and ideas. Being able to reflect on one's own actions and thoughts is a hallmark of the most sophisticated forms of artificial intelligence. The problem is that this kind of AI has not yet progressed beyond the theoretical stage.

### **8.1. AI and web development**

Web developers, programmers, and engineers are finding ChatGPT and other AI tools to be incredibly helpful as of the spring of 2023, and these technologies are rapidly changing the way websites are planned, produced, and maintained. Key areas of AI's impact on web development include:

Web Design – A wide range of processes are being automated and enhanced using AI, including:

- Design Generation – The utilisation of user input or predetermined templates allows AI-powered tools to develop design layouts. In order to shorten the design process, these technologies study customer preferences and generate unique design suggestions.
- Color Scheme Selection – Through the analysis of photos, AI is able to produce colour schemes that align with the visual aesthetics of a concept or brand. In doing so, it is easier to keep the overall style and aesthetic of a website consistent.
- Typography – Tools powered by AI can analyse a design and its content to determine the best typographic styles and font combinations, guaranteeing a top-notch user experience.
- Web Development – Applications powered by AI are currently changing the game when it comes to website development and management. These innovations include:
- Code Generation and Coding Assistant – With the help of AI, web developers may have code generated automatically from either user input or predetermined templates. Because of this, the development process is accelerated and human mistake is decreased. Code snippets or even complete apps can be generated in real time with its support and assistance for programmers. Code refactoring is also possible with it.
- Automated Testing and Problem-Finding with Bug Fixes – Artificial intelligence may mimic user movements like clicking and swiping to evaluate how well a website works, how responsive it is, and how fast it loads. Because of this, developers can find and solve errors faster, which means the user experience is smooth. To top it all off, it can detect and recommend fixes for syntax and logic issues.
- Code Documentation – Artificial intelligence is capable of accurately documenting and describing the behaviour of any piece of code. Code comments are another capability.

Numerous products are in the process of being released, including Microsoft's Copilot X for GitHub, CodeGPT for Visual Studio Code (VSC), and ChatGPT from OpenAI. Optimisation of Content—Tools powered by artificial intelligence can facilitate optimisation, management, and development of content, including:

- Natural Language Processing – Tools such as content summarisers, sentiment analysis, and keyword extraction are made possible by AI's ability to comprehend and analyse human language. This allows for the optimisation of material for both search engines and user engagement.
- Content Generation – With the help of AI, articles and social media postings can be generated according to particular themes or keywords. This can guarantee continuous content quality while saving time and resources.
- Image and Video Optimization and creation – Images and videos can be automatically resized, compressed, and enhanced for web use using algorithms powered by artificial intelligence (AI) without sacrificing quality. It may also make new films, including short clips in different formats, and pictures, including icons, photos, animations, etc.

Tools like ChatGPT and OpenAI's Midjourney are being released at the moment.

### **8.2. Generative Pre-trained Transformer – GPT**

The research foundation OpenAI created an advanced AI language model called GPT (Generative Pre-trained Transformer) with a primary focus on tasks related to natural language understanding and generation. The model is becoming more complicated as the amount of data utilised for training increases. This is explicable by:

**Transformer Architecture** – The architecture of the transformer forms the basis. In place of more conventional neural networks like convolutional or recurrent, transformers use a self-attention mechanism to process sequential data like text. A variety of language-focused tasks are improved by the model's ability to use this self-attention mechanism to prioritise different parts of the input sequence when producing predictions.

**Pre-training and Fine-tuning** – It uses a two-stage procedure that involves pre-training and fine-tuning. To prepare for training, the model is fed massive volumes of text and taught to infer the meaning of the following word in a phrase from the context of the one before it. Gradually, GPT is able to learn grammar, syntax, and even some general understanding of the world because to its unsupervised learning. To execute language-specific tasks like sentiment analysis, summarisation, or question-answering, the model undergoes further training on task-specific labelled data during the fine-tuning phase.

**Versions of GPT** – By 2023, four significant revisions of GPT will have been released:

- **GPT** – In 2018, the initial GPT model showed that the transformer architecture and the pre-training/fine-tuning method worked well for different language-specific tasks.
- **GPT-2** – Released in 2019, GPT-2 significantly enhanced language interpretation and generating skills through training on a larger dataset and boosting the model size to 1.5 billion parameters. Several benchmarks demonstrated that GPT2 could produce coherent and contextually appropriate text, and it did so with state-of-the-art performance.
- **GPT-3** – Released in 2020, GPT-3 is an advanced version that has been trained on an even bigger dataset and features 175 billion parameters. Despite requiring little in the way of fine-tuning, GPT-3 shows outstanding performance on a number of language-specific tasks, and it has been extensively used in commercial settings. When ChatGPT's popularity skyrocketed a few months ago, with 100 million users in only two months, GPT-3.5 (Turbo) was the breakthrough version for the greater public.

### **8.3. Computer languages and its tools**

Humans and machines, as well as humans and the web, are able to communicate thanks to the basic building blocks of communication: computer and web languages. Web applications, software, and interactive user interfaces can be built using these programming languages, which have a predetermined syntax and ruleset.

### **8.4. Hypertext Preprocessor – PHP**

PHP (Hypertext Preprocessor) is an open-source scripting language for servers that was created specifically for web development and is used by many. In 1994, Rasmus Lerdorf made it. After PHP runs on the server, the client receives plain HTML as a result. Form data processing, session management, and database access are all areas where it shines because of this. PHP is widely compatible with HTML and may be used in conjunction with a number of web frameworks and content management systems, such as Laravel.

**JavaScript** – JS JS (JavaScript) is an integral part of contemporary web development and is a flexible, high-level, interpreted programming language. In 1995, Brendan Eich created JavaScript, which is now standardised as ECMAScript. Its purpose is to make webpages more interactive and to add dynamic content.

In contrast to PHP, which operates on the server side, JavaScript is mainly used by the user in their browser. With this, users can enjoy better experiences including real-time updates, animations, and event handling—all without constantly communicating with the server. [15] Now that it supports server-side development with technologies like Node, JavaScript can be considered a full-stack language. React, Angular, Vue, AJAX, and jQuery are just a few of the web frameworks and libraries that rely on it to facilitate web development and encourage modular, reusable code.

### **8.5. Content Management Systems – CMS**

Users without in-depth technical understanding can produce, manage, and publish digital material on websites with the help of a material Management System (CMS). Websites with a professional appearance may be easily created and maintained by anyone with an interest thanks to content management systems that streamline the process of traditional web development. As a result of its intuitive UI (User Interface), users are free to concentrate on the content and essential data rather than the underlying coding, which facilitates both content production and management. Customisable themes and templates, along with a wide range of plugins and extensions, are standard features of most content management systems (CMS). There are a number of prominent content management system (CMS) systems out there; three of the most notable are WordPress, Joomla, and Drupal. Websites ranging from simple blog sites and small

businesses to elaborate e-commerce platforms and enterprise-level solutions may all be created with content management systems.

WordPress – WP

The most widely used open-source content management system (CMS) for building and maintaining websites is WP (WordPress). Since its 2003 launch by Matt Mullenweg and Mike Little, it has grown to become the most popular content management system (CMS) in the world, supporting more than 40% of all websites. From its humble beginnings as a blogging platform, WP has expanded to include many more kinds of websites, such as portfolio sites, e-commerce businesses, and enterprise-level solutions. Its user-friendly interface (UI) and extensive library of themes and plugins make it suitable for users with varying degrees of experience. The vast ecology that WP offers is one of its main advantages. Users can alter the look and functionality of their websites with the help of thousands of plugins and themes, some of which are paid and others of which are free.

### 8.6. Application Programming Interface – API

An API, or Application Programming Interface, is a collection of standards, protocols, and tools that facilitate communication across various software and online applications. For example, an API could enable a web service to retrieve data from an external database hosted on a separate server. Without understanding how those external systems are structured, developers can use these interfaces to obtain the data or features they need. Typically, developers merely require knowledge of the URL routes for all data and functionality. Web, operating system, and library APIs are only a few examples of the many kinds of application programming interfaces (APIs). In the age of web services and cloud computing, web APIs have grown in significance. The CRUD process—which stands for "create," "retrieve," "update," and "delete"—is the main way in which web APIs facilitate communication between various devices. In addition, a RESTful API (REpresentational State Transfer) places greater emphasis on the ways in which web services make use of APIs to execute HTTP operations, such as retrieving resources through base URL encoding parameters and transferring data in JSON or XML format. The CRUD principles are often followed by the actions. The goal of REST is to make it easier for various parties to handle data.

---

## 9. Conclusion

Using CodeGPT brings back a flood of memories of how my classes introduced me to project management, web development, and coding from the ground up. For instance, GPT mostly teaches by describing the processes to build an application and how to utilise a single piece of software, which is the ideal approach to teaching programming from an educational standpoint. Furthermore, it is capable of providing the coder with factually accurate, instructional, and extensive explanations of every step or section of code that they require. Without a doubt, this type of instant learning boosts a student's self-confidence and provides information to move on to the next step, as opposed to spending a long time searching for answers on Google, which interrupts the thinking process. But CodeGPT has its flaws. The top instructors incorporate visual aids and graphics into their lessons, which CodeGPT and ChatGPT lack. This helps students understand the topic much more quickly and effectively. When compared to other course materials, this one has better use of visual aids. Though it is still in its prototype stage for PowerPoint, Word, and Excel, I anticipate that the GPT-5 model will include a small amount of picture processing and generation when it comes to tables and diagrams. This functionality is now available in third-party solutions.

---

## Compliance with ethical standards

*Disclosure of conflict of interest*

No conflict of interest to be disclosed.

---

## References

- [1] Introduction to the Angular Docs. Available online: <https://angular.io/docs> (accessed on 20 July 2023).
- [2] Sanderson, K. GPT-4 is here: What scientists think. *Nature* 2023, 615, 773. [Google Scholar] [CrossRef]
- [3] Deng, J.; Lin, Y. The Benefits and Challenges of ChatGPT: An Overview. *Front. Comput. Intell. Syst.* 2023, 2, 81–83. [Google Scholar] [CrossRef]

- [4] Jánki, Z.R.; Bilicki, V. Rule-Based Architectural Design Pattern Recognition with GPT Models. *Electronics* 2023, 12, 3364. [Google Scholar] [CrossRef]
- [5] Hourani, H.; Hammad, A.; Lafi, M. The Impact of Artificial Intelligence on Software Testing. In *Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, Amman, Jordan, 9–11 April 2019. [Google Scholar] [CrossRef]
- [6] Heydon, A.; Maimone, M.; Tygar, J.; Wing, J.; Zaremski, A. Miro: Visual specification of security. *IEEE Trans. Softw. Eng.* 1990, 16, 1185–1197. [Google Scholar] [CrossRef]
- [7] Giordano, M.; Polese, G. Visual Computer-Managed Security: A Framework for Developing Access Control in Enterprise Applications. *IEEE Softw.* 2013, 30, 62–69. [Google Scholar] [CrossRef]
- [8] Hossain Misu, M.R.; Sakib, K. FANTASIA: A Tool for Automatically Identifying Inconsistency in AngularJS MVC Applications. In *Proceedings of the Twelfth International Conference on Software Engineering Advances*, Athens, Greece, 8–12 October 2017. [Google Scholar]
- [9] Szabó, Z.; Bilicki, V. Access Control of EHR Records in a Heterogeneous Cloud Infrastructure. *Acta Cybern.* 2021, 25, 485–516. [Google Scholar] [CrossRef]
- [10] Martin, B.; Brown, M.; Paller, A.; Kirby, D.; Christey, S. *CWE. SANS Top 2011*, 25. [Google Scholar]
- [11] Rainey, S.; McGillivray, K.; Akintoye, S.; Fothergill, T.; Bublitz, C.; Stahl, B. Is the European Data Protection Regulation sufficient to deal with emerging data concerns relating to neurotechnology? *J. Law Biosci.* 2020, 7, lsa051. [Google Scholar] [CrossRef]
- [12] Cheng, S.; Zhang, J.; Dong, Y. How to Understand Data Sensitivity? A Systematic Review by Comparing Four Domains. In *Proceedings of the 2022 4th International Conference on Big Data Engineering*, Beijing, China, 26–28 May 2022. [Google Scholar] [CrossRef]
- [13] Belen Saglam, R.; Nurse, J.R.; Hodges, D. Personal information: Perceptions, types and evolution. *J. Inf. Secur. Appl.* 2022, 66, 103163. [Google Scholar] [CrossRef]
- [14] Lang, C.; Woo, C.; Sinclair, J. Quantifying data sensitivity. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, Frankfurt, Germany, 23–27 March 2020. [Google Scholar] [CrossRef]
- [15] Chua, H.N.; Ooi, J.S.; Herbland, A. The effects of different personal data categories on information privacy concern and disclosure. *Comput. Secur.* 2021, 110, 102453. [Google Scholar] [CrossRef]
- [16] Rumbold, J.M.; Pierscionek, B.K. What Are Data? A Categorization of the Data Sensitivity Spectrum. *Big Data Res.* 2018, 12, 49–59.